Discontinuous Galerkin methods for compressible flows: higher order accuracy, error estimation and adaptivity

Ralf Hartmann
Institute of Aerodynamics and Flow Technology
German Aerospace Center (DLR)
Lilienthalplatz 7, 38108 Braunschweig, Germany

Contents

1 Introduction

In recent years discontinuous Galerkin methods have experienced a resurgence of interest in multivarious disciplines of numerical mathematics including compressible flows and aerodynamics, [6, 7, 9, 10, 16, 22, 23, 24, 36], among many others. It can be observed that to an increasing extent discontinuous Galerkin methods are now applied to problems which traditionally where solved using finite volume methods. The reason for this trend can be identified in several advantages of the discontinuous Galerkin methods over finite volume methods. Second order finite volume methods are achieved by employing a second order accurate reconstruction. The extension of a second order finite volume scheme to a (theoretically) third order scheme requires a third order accurate reconstruction which on unstructured meshes is very cumbersome and which in practice shows deterioration of order. On unstructured meshes finite volume methods of even higher order are virtually impossible. These difficulties bound the order of numerical computations in industrial applications to second order. In contrast to this, the order of discontinuous Galerkin methods, applied to problems with regular solutions, depends on the degree of the approximating polynomials only which can easily be increased, dramatically simplifying the use of higher order methods on unstructured meshes. Furthermore, the stencil of most discontinuous Galerkin schemes is minimal in the sense that each element communicates only with its direct neighbours. In contrast to the increasing number of elements or mesh points communicating for increasing accuracy of finite volume methods, the inter-element communication of discontinuous Galerkin methods is the same for any order. The compactness of the discontinuous Galerkin method has clear advantages in parallelisation, which does not require additional element layers at partition boundaries. Also due to simple communication at element interfaces, elements with so-called 'hanging nodes' can be treated just as easily as elements without hanging nodes, a fact that simplifies local mesh refinement (h-refinement). In addition to this, the communication at element interfaces is identical for any order of the method which simplifies the use of methods of differing orders in adjacent elements. This allows for the variation of the order of the numerical scheme over the computational domain, which in combination with h-refinement leads to so-called hp-refinement, where p-refinement denotes the variation of the polynomial degree p.

This article starts with providing the basics of discontinuous Galerkin (DG) methods in Section 2. The DG method for hyperbolic problems is then extended to the compressible Navier-Stokes equations of gasdynamics in Section 4.2. Higher order computations for inviscid and viscous flows are shown in Section 3 and 6, respectively. Further topics covered are: The Newton-GMRES algorithm for the solution of the nonlinear equations including a detailed description of Jacobians in Section 5, the stabilization of the numerical scheme with shock-capturing in Section 7 and finally the definition of adjoint problems and their use in error estimation and adaptive mesh refinement in Section 8.

2 The Discontinuous Galerkin discretization of hyperbolic conservation equations

2.1 Hyperbolic conservation equations

Given a final time $T > 0$, we consider the following system of conservation equations,

$$\begin{aligned} \frac{\partial}{\partial t}\mathbf{u} + \sum_{i=1}^{d} \frac{\partial}{\partial x_i}\mathbf{f}_i^c(\mathbf{u}) = 0 \qquad & \text{in } (0,T]\times\Omega, \\ \mathbf{u}(0,\cdot) = \mathbf{u}_0(\cdot) \quad & \text{in } \Omega, \end{aligned} \tag{1}$$

where Ω is a bounded connected domain in $\mathbb{R}^d$, $d \geq 1$, $\mathbf{u} = (u_1, \ldots, u_m)^T$, $\mathcal{F}^c(\mathbf{u}) = (\mathbf{f}_1^c(\mathbf{u}), \ldots, \mathbf{f}_d^c(\mathbf{u}))$ and $\mathbf{f}_i^c : \mathbb{R}^m \to \mathbb{R}^m$, $i = 1, \ldots, d$, are continuously differentiable. In particular, we will be concerned with the solution of the *stationary* system of conservation equations,

$$\nabla \cdot \mathcal{F}^c(\mathbf{u}) = 0 \quad \text{in } \Omega, \tag{2}$$

subject to appropriate boundary conditions described below. We say that (1) is hyperbolic, if the matrix

$$B(\mathbf{u}, \boldsymbol{\nu}) := \sum_{i=1}^{d} \nu_i A_i(\mathbf{u}) \tag{3}$$

has m real eigenvalues and a complete set of linearly independent eigenvectors for all vectors $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_d) \in \mathbb{R}^d$. Here, $A_i(\mathbf{u}) \in \mathbb{R}^{m\times m}$ denotes the Jacobi matrix of the flux $\mathbf{f}_i^c(u)$, i.e.

$$A_i(\mathbf{u}) := \frac{\partial}{\partial \mathbf{u}}\mathbf{f}_i^c(\mathbf{u}), \qquad i = 1, \ldots, d. \tag{4}$$

The system of conservation equations (1) must be supplemented with appropriate boundary conditions; for example at inflow/outflow boundaries, we require that

$$B^-(\mathbf{u}, \mathbf{n})\,(\mathbf{u} - \mathbf{g}) = \mathbf{0}, \quad \text{on } \Gamma \tag{5}$$

where $\mathbf{n}$ denotes the unit outward normal vector to the boundary $\Gamma = \partial\Omega$ and $\mathbf{g}$ is a (given) vector function. Here, $B^{\pm}(\mathbf{u}, \mathbf{n})$ denotes the negative/positive part of $B(\mathbf{u}, \mathbf{n})$,

$$B^{\pm}(\mathbf{u}, \mathbf{n}) = P\Lambda^{\pm}P^{-1}, \tag{6}$$

where $P = [\mathbf{r}_1, \ldots, \mathbf{r}_m]$ denotes the $m \times m$ matrix of eigenvectors of $B(\mathbf{u}, \mathbf{n})$ and $\Lambda^- = \mathrm{diag}(\min(\lambda_i, 0))$ and $\Lambda^+ = \mathrm{diag}(\max(\lambda_i, 0))$ the $m \times m$ diagonal matrix of the negative/positive eigenvalues of $B(\mathbf{u}, \mathbf{n})$ with $B\mathbf{r}_i = \lambda_i \mathbf{r}_i$, $i = 1, \ldots, d$.

In the following, we give two examples of hyperbolic conservation equations.

2.2 Scalar hyperbolic problem

The linear advection equation is given by

$$\begin{aligned} \boldsymbol{\beta} \cdot \nabla u = f \quad & \text{in } \Omega, \\ u = g \quad & \text{on } \Gamma_-, \end{aligned} \tag{7}$$

where $\boldsymbol{\beta} \in \mathbb{R}^d$ and $\Gamma_- \subset \Gamma$ denotes the inflow boundary defined by

$$\Gamma_- = \{\mathbf{x} \in \Gamma, (\boldsymbol{\beta} \cdot \mathbf{n})(\mathbf{x}) < 0\}.$$

This is the simplest example of a hyperbolic problem as given in (2). In fact, setting $\mathcal{F}^c(u) = (\beta_1 u, \ldots, \beta_d u) = \boldsymbol{\beta} u$, with a scalar function u, i.e. $m = 1$, we have $B(u, \mathbf{n}) = \boldsymbol{\beta} \cdot \mathbf{n} = \lambda \in \mathbb{R}$. The boundary condition (5) is given by

$$B^-(u, \mathbf{n})(u - g) = \lambda^-(u - g) = 0 \quad \text{on } \Gamma,$$

where $\lambda^- = \min(\lambda, 0)$ which reduces to $u = g$ on Γ_- and no boundary condition is imposed on the outflow boundary $\Gamma_+ = \{\mathbf{x} \in \Gamma, (\boldsymbol{\beta} \cdot \mathbf{n})(\mathbf{x}) \geq 0\}$.

2.3 The compressible Euler equations

The Euler equations of compressible gas dynamics represent an important example of the hyperbolic problem (1). In two space–dimensions, the vector of conservative variables $\mathbf{u}$ and the convective fluxes $\mathbf{f}_i^c$, $i = 1, 2$, are defined by

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho v_1 \\ \rho v_2 \\ \rho E \end{bmatrix}, \quad \mathbf{f}_1^c(\mathbf{u}) = \begin{bmatrix} \rho v_1 \\ \rho v_1^2 + p \\ \rho v_1 v_2 \\ \rho H v_1 \end{bmatrix} \text{ and } \mathbf{f}_2^c(\mathbf{u}) = \begin{bmatrix} \rho v_2 \\ \rho v_1 v_2 \\ \rho v_2^2 + p \\ \rho H v_2 \end{bmatrix}, \tag{8}$$

where ρ, $\mathbf{v} = (v_1, v_2)^T$, p and E denote the density, velocity vector, pressure and specific total energy, respectively. Additionally, H is the total enthalpy given by

$$H = E + \frac{p}{\rho} = e + \tfrac{1}{2}\mathbf{v}^2 + \frac{p}{\rho}, \tag{9}$$

where e is the specific static internal energy, and the pressure is determined by the equation of state of an ideal gas

$$p = (\gamma - 1)\rho e, \tag{10}$$

where $\gamma = c_p/c_v$ is the ratio of specific heat capacities at constant pressure, c_p, and constant volume, c_v; for dry air, $\gamma = 1.4$. The flux Jacobians $A_i(\mathbf{u})$ defined in (4) are given by

$$A_1(\mathbf{u}) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -v_1^2 + \frac{1}{2}(\gamma - 1)\mathbf{v}^2 & (3 - \gamma)v_1 & -(\gamma - 1)v_2 & \gamma - 1 \\ -v_1 v_2 & v_2 & v_1 & 0 \\ v_1\left(\frac{1}{2}(\gamma - 1)\mathbf{v}^2 - H\right) & H - (\gamma - 1)v_1^2 & -(\gamma - 1)v_1 v_2 & \gamma v_1 \end{pmatrix},$$

$$A_2(\mathbf{u}) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ -v_1 v_2 & v_2 & v_1 & 0 \\ -v_2^2 + \frac{1}{2}(\gamma - 1)\mathbf{v}^2 & -(\gamma - 1)v_1 & (3 - \gamma)v_2 & \gamma - 1 \\ v_2\left(\frac{1}{2}(\gamma - 1)\mathbf{v}^2 - H\right) & -(\gamma - 1)v_1 v_2 & H - (\gamma - 1)v_2^2 & \gamma v_2 \end{pmatrix}.$$

Finally, the eigenvalues of the matrix $B(\mathbf{u}, \mathbf{n}) = \sum_{i=1}^2 n_i A_i(\mathbf{u})$ are

$$\lambda_1 = \mathbf{v} \cdot \mathbf{n} - c, \quad \lambda_2 = \lambda_3 = \mathbf{v} \cdot \mathbf{n}, \quad \lambda_4 = \mathbf{v} \cdot \mathbf{n} + c \tag{11}$$

where $c = \sqrt{\gamma p/\rho}$ denotes the speed of sound. Considering the signs of λ_i, $i = 1, \ldots, 4$, we distinguish four cases of boundary conditions (5):

- supersonic inflow: $\lambda_i < 0$, $i = 1, \ldots, 4$,

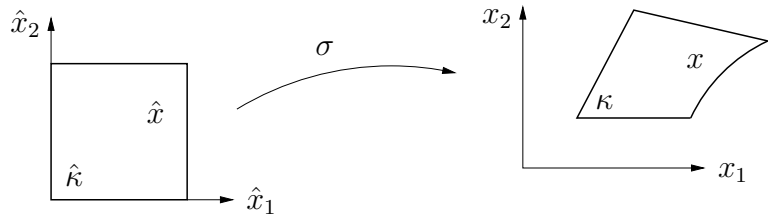


Figure 1: Mapping σ of reference element $\hat{\kappa}$ to the element κ in real space.

- subsonic inflow: $\lambda_i < 0$, $i = 1, 2, 3$, $\lambda_4 > 0$,
- subsonic outflow: $\lambda_1 < 0$, $\lambda_i > 0$, $i = 2, 3, 4$, and
- supersonic outflow: $\lambda_i > 0$, $i = 1, \ldots, 4$.

Each eigenvalue smaller than zero corresponds to an inflow characteristic. The number of variables to be prescribed on the boundary depends on the number of inflow characteristics.

2.4 The Discontinuous Galerkin discretization

In this section we state the discontinuous Galerkin discretization of hyperbolic conservation equations (2).

First, we begin by introducing some notation. We assume that Ω can be subdivided into shape-regular meshes $\mathcal{T}_h = \{\kappa\}$ consisting of quadrilateral elements κ. Here, h denotes the piecewise constant mesh function defined by $h|_\kappa \equiv h_\kappa = \mathrm{diam}(\kappa)$ for all $\kappa \in \mathcal{T}_h$. Let us assume that each $\kappa \in \mathcal{T}_h$ is an image of a fixed reference element $\hat{\kappa}$, that is, $\kappa = \sigma_\kappa(\hat{\kappa})$ for all $\kappa \in \mathcal{T}_h$. Here, we shall only consider the case when $\hat{\kappa}$ is the open unit hypercube in $\mathbb{R}^d$. Furthermore the mapping σ_κ of the reference element $\hat{\kappa}$ to the element κ in real space is assumed to be bijective and smooth, with the eigenvalues of its Jacobian matrix being bounded from below and above. For elements in the interior of the domain, $\partial\kappa \cap \Gamma = \emptyset$, the mapping σ_κ is given by a d-linear function; In order to represent curved boundaries, see Figure 1, mappings can be used that include polynomials of higher degree on boundary elements, see [19] for more details about curved elements.

On the reference element $\hat{\kappa}$ we define spaces of tensor product polynomials of degree $p \geq 0$ as follows:

$$\mathcal{Q}_p(\hat{\kappa}) = \mathrm{span}\left\{\hat{x}^\alpha : 0 \leq \alpha_i \leq p,\ 0 \leq i \leq d\right\}, \tag{12}$$

where α denotes a multi-index and $\hat{x}^\alpha = \prod_{i=1}^d \hat{x}_i^{\alpha_i}$. Finally, we introduce the finite element space $\mathbf{V}_h^p$ consisting of discontinuous vector–valued polynomial functions of degree $p \geq 0$, defined by

$$\mathbf{V}_h^p = \{\mathbf{v}_h \in [L_2(\Omega)]^m : \mathbf{v}_h|_\kappa \circ \sigma_\kappa \in [\mathcal{Q}_p(\hat{\kappa})]^m\}. \tag{13}$$

Suppose that $\mathbf{v}|_\kappa \in [H^1(\kappa)]^m$ for each $\kappa \in \mathcal{T}_h$. Let κ and κ' be two adjacent elements of $\mathcal{T}_h$ and $\mathbf{x}$ be an arbitrary point on the interior edge $e = \partial\kappa \cap \partial\kappa'$. By $\mathbf{v}_\kappa^\pm$ we denote the traces of $\mathbf{v}$ taken from within the interior of κ and κ', respectively, see Figure 2. For $\mathbf{x} \in \partial\kappa \cap \Gamma$ the outer trace is set to be $\mathbf{v}_\kappa^- := \mathbf{g}$ where $\mathbf{g}$ denotes an appropriate boundary function.

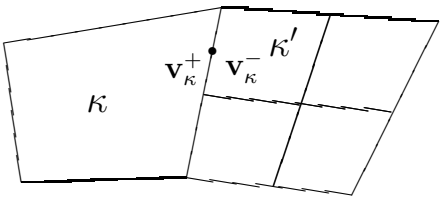


Figure 2: Definition of the interior and outer traces $\mathbf{v}_\kappa^\pm$ wrt. element κ.

Since below it will always be clear from the context which element κ in the subdivision $\mathcal{T}_h$ the quantities $\mathbf{v}_\kappa^+$ and $\mathbf{v}_\kappa^-$ correspond to, for the sake of notational simplicity, we shall suppress the letter κ in the subscript and write, respectively, $\mathbf{v}^+$ and $\mathbf{v}^-$, instead.

To formulate the discontinuous Galerkin method, we first introduce a weak formulation of (2). To this end, we multiply the equation (2) by an arbitrary smooth (vector-)function $\mathbf{v}$ and integrate by parts over an element κ in the mesh $\mathcal{T}_h$; thereby, we obtain

$$-\int_\kappa \mathcal{F}^c(\mathbf{u}) \cdot \nabla \mathbf{v} \, \mathrm{d}\mathbf{x} + \int_{\partial\kappa} \mathcal{F}^c(\mathbf{u}) \cdot \mathbf{n} \, \mathbf{v} \, ds = 0. \tag{14}$$

To discretize (14), we replace the analytical solution $\mathbf{u}$ by the Galerkin finite element approximation $\mathbf{u}_h$ and the test function $\mathbf{v}$ by $\mathbf{v}_h$, where $\mathbf{u}_h$ and $\mathbf{v}_h$ both belong to the finite element space $\mathbf{V}_h^p$. In addition, since the numerical solution $\mathbf{u}_h$ is discontinuous between element interfaces, we must replace the flux $\mathcal{F}^c(\mathbf{u})\cdot\mathbf{n}$ by a *numerical flux* function $\mathcal{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n})$, which depends on both the interior– and outer–trace of $\mathbf{u}_h$ on $\partial\kappa$, $\kappa \in \mathcal{T}_h$, and the unit outward normal $\mathbf{n}$ to $\partial\kappa$. Thereby, summing over the elements κ in the mesh $\mathcal{T}_h$, yields the discontinuous Galerkin finite element discretization of (2) as follows: find $\mathbf{u}_h \in \mathbf{V}_h^p$ such that

$$\sum_{\kappa\in\mathcal{T}_h} \left\{ -\int_\kappa \mathcal{F}^c(\mathbf{u}_h) \cdot \nabla \mathbf{v}_h \, \mathrm{d}\mathbf{x} + \int_{\partial\kappa} \mathcal{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n}) \, \mathbf{v}_h^+ \, ds \right\} = 0 \quad \forall \mathbf{v}_h \in \mathbf{V}_h^p. \tag{15}$$

This scheme is called discontinuous Galerkin method of degree p, or in short, "DG(p) method".

For elements $\kappa \in \mathcal{T}_h$ whose boundaries intersect that of the computational domain Ω, we replace $\mathbf{u}_h^-$ by appropriate boundary conditions on the portion of $\partial\kappa$ for which $\partial\kappa \cap \Gamma \neq \emptyset$. For more details about boundary conditions for the compressible Euler and Navier-Stokes equations, see Section 4.2.

We remark that the replacement of the flux $\mathcal{F}^c(\mathbf{u}) \cdot \mathbf{n}$ by the numerical flux function $\mathcal{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n})$ on the boundary of element κ, κ in $\mathcal{T}_h$, corresponds to the weak imposition of the boundary data. Furthermore, we emphasize that the choice of the numerical flux function is *independent* of the finite element space employed. Indeed, the numerical flux $\mathcal{H}(\cdot,\cdot,\cdot)$ may be chosen to be any two–point monotone Lipschitz function which satisfies the following two conditions:

(i) $\mathcal{H}(\cdot,\cdot,\cdot)|_{\partial\kappa}$ is consistent with the flux $\mathcal{F}^c(\cdot)\cdot\mathbf{n}$ for each κ in $\mathcal{T}_h$; i.e.

$$\mathcal{H}(\mathbf{v},\mathbf{v},\mathbf{n})|_{\partial\kappa} = \mathcal{F}^c(\mathbf{v})\cdot\mathbf{n} \quad \forall \kappa \in \mathcal{T}_h;$$

(ii) $\mathcal{H}(\cdot,\cdot,\cdot)$ is conservative, i.e. given any two neighbouring elements κ and κ' from the finite element partition $\mathcal{T}_h$, at each point $x \in \partial\kappa \cap \partial\kappa' \neq \emptyset$, noting that $\mathbf{n}_{\kappa'} = -\mathbf{n}$, we have that

$$\mathcal{H}(\mathbf{v},\mathbf{w},\mathbf{n}) = -\mathcal{H}(\mathbf{w},\mathbf{v},-\mathbf{n}).$$

There are several numerical flux functions satisfying these conditions, such as the Godunov, Engquist–Osher, Lax–Friedrichs, Roe or the Vijayasundaram flux, for example; cf. [28, 35] and the references cited therein. As examples, here we consider two different numerical fluxes: the (local) Lax–Friedrichs flux and the Vijayasundaram flux.

The **(local) Lax–Friedrichs flux** $\mathcal{H}_{LF}(\cdot,\cdot,\cdot)$, is defined by

$$\mathcal{H}_{LF}(\mathbf{u}^+,\mathbf{u}^-,\mathbf{n})|_{\partial\kappa} = \tfrac{1}{2}\left(\mathcal{F}^c(\mathbf{u}^+)\cdot\mathbf{n} + \mathcal{F}^c(\mathbf{u}^-)\cdot\mathbf{n} + \alpha\left(\mathbf{u}^+ - \mathbf{u}^-\right)\right), \tag{16}$$

for $\kappa \in \mathcal{T}_h$, where α is the maximum over $\mathbf{u}^+$ and $\mathbf{u}^-$,

$$\alpha = \alpha\left(\mathbf{u}^+,\mathbf{u}^-\right) = \max_{\mathbf{v}=\mathbf{u}^+,\mathbf{u}^-}\{|\lambda(B(\mathbf{v},\mathbf{n}))|\}, \tag{17}$$

of the largest eigenvalue (in absolute value) $|\lambda(B)|$ of the matrix $B(\mathbf{v},\mathbf{n}) = \sum_{i=0}^{d} n_i A_i(\mathbf{u})$ defined in (3).

The **Vijayasundaram flux** $\mathcal{H}_V(\cdot,\cdot,\cdot)$, is defined by

$$\mathcal{H}_V(\mathbf{u}^+,\mathbf{u}^-,\mathbf{n})|_{\partial\kappa} = B^+(\bar{\mathbf{u}},\mathbf{n})\mathbf{u}^+ + B^-(\bar{\mathbf{u}},\mathbf{n})\mathbf{u}^- \quad \text{for } \kappa \in \mathcal{T}_h, \tag{18}$$

where $B^+(\bar{\mathbf{u}},\mathbf{n})$ and $B^-(\bar{\mathbf{u}},\mathbf{n})$ denote the positive and negative parts, cf. (6), of the matrix $B(\bar{\mathbf{u}},\mathbf{n})$, respectively, evaluated at an average state $\bar{\mathbf{u}}$ between $\mathbf{u}^+$ and $\mathbf{u}^-$.

Remark. We note that the discontinuous Galerkin discretization is similar to finite Volume schemes, in particular in the use of numerical flux functions. In fact, the basic finite Volume scheme corresponds to the DG(0) method, i.e. to the discontinuous Galerkin method using piecewise constant ansatz and test functions. Consequently, the DG(p) methods with $p > 0$ can be considered as the "natural" generalization of finite Volume methods to higher order methods. Indeed, simply by increasing the degree p of the discrete function space $\mathbf{V}_h^p$ in the discretization (15) we gain discontinuous Galerkin schemes of corresponding higher orders. This will be demonstrated by a numerical example of the two-dimensional compressible Euler equations in Section 3.

2.5 DG for scalar hyperbolic problems

Originally, the discontinuous Galerkin method was introduced by Reed and Hill [31] and analysed by LeSaint and Raviart [29] for following scalar hyperbolic problem used as a simple model for neutron transport

$$\begin{aligned} \boldsymbol{\beta}\cdot\nabla u + bu &= f \quad \text{in } \Omega, \\ u &= g \quad \text{on } \Gamma_-, \end{aligned} \tag{19}$$

with constant vector $\boldsymbol{\beta} \in R^2$ and a real number $b > 0$. The discontinuous Galerkin discretization of this problem is given by

$$\sum_{\kappa\in\mathcal{T}_h}\left\{\int_\kappa (\boldsymbol{\beta}\cdot\nabla u_h + bu_h)v_h\,\mathrm{d}\mathbf{x} - \int_{\partial\kappa_-}\boldsymbol{\beta}\cdot\mathbf{n}\,[u_h]v_h^+\,\mathrm{d}s\right\} = \sum_{\kappa\in\mathcal{T}_h}\left\{\int_\kappa f v_h\,\mathrm{d}\mathbf{x}\right\}, \tag{20}$$

with the jump $[u_h] = u_h^+ - u_h^-$, the inflow boundary of the element,

$$\partial\kappa_- = \{x \in \partial\kappa, \boldsymbol{\beta}\cdot n < 0\},$$

the outflow boundary $\partial\kappa_+ = \Gamma \setminus \partial\kappa_-$, and inflow boundary values

$$u_h^-(\mathbf{x}) = g(\mathbf{x}), \quad \mathbf{x} \in \Gamma_-,$$

see also Johnson and Pitkäranta [27], for example. At first sight, the discontinuous Galerkin discretization given by (15) looks different to the original DG discretization (20). However, by defining the numerical flux for the scalar hyperbolic problem to be

$$\mathcal{H}(u_h^+, u_h^-, \mathbf{n})(\mathbf{x}) = \begin{cases} \boldsymbol{\beta} \cdot \mathbf{n}\, u_h^-, & \text{for } (\boldsymbol{\beta} \cdot \mathbf{n})(\mathbf{x}) < 0, \text{ i.e. } \mathbf{x} \in \partial\kappa_-, \\ \boldsymbol{\beta} \cdot \mathbf{n}\, u_h^+, & \text{for } (\boldsymbol{\beta} \cdot \mathbf{n})(\mathbf{x}) \geq 0, \text{ i.e. } \mathbf{x} \in \partial\kappa_+, \end{cases} \tag{21}$$

we see that after integration by parts equation (15) reduces to (20) with $b = 0$ and $f = 0$. We recall that the numerical flux replaces the flux term $\mathcal{F}^c(u) \cdot \mathbf{n}$ on the boundary of the elements. The normal flux of the scalar hyperbolic conservation equation is given by $\mathcal{F}^c(u) \cdot \mathbf{n} = \boldsymbol{\beta} \cdot \mathbf{n}\, u$, and the numerical flux $\mathcal{H}$ defined in (21), switches between the interior and the outer traces, u_h^+ and u_h^-, respectively, depending on $\mathbf{x} \in \partial\kappa$ being on the outflow or the inflow part of the boundary of the element. Furthermore, we note, that both, the local Lax–Friedrichs flux (16) and the Vijayasundaram flux (18), reduce to the flux given in (21) when applied to the scalar hyperbolic problem (19).

2.6 DG for scalar hyperbolic problems: *A priori* error estimates

In the following we quote an *a priori* error estimate for the discontinuous Galerkin method of the scalar hyperbolic problem (19):

Let $b \in C(\bar{\Omega})$, $f \in L^2(\Omega)$, and $g \in L^2(\Gamma_-)$ on the inflow boundary $\Gamma_- := \{x \in \Gamma, \boldsymbol{\beta}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) < 0\}$, then the weak formulation of (19) given by,

$$\begin{aligned} \int_\Omega (\boldsymbol{\beta} \cdot \nabla u + bu) v \, d\mathbf{x} &= \int_\Omega f v \, d\mathbf{x}, \qquad v \in L^2(\Omega), \\ \int_{\Gamma_-} \boldsymbol{\beta} \cdot \mathbf{n}\, uw \, ds &= \int_{\Gamma_-} \boldsymbol{\beta} \cdot \mathbf{n}\, gw \, ds, \qquad w \in L^2(\Gamma_-), \end{aligned}$$

has a unique (weak) solution $u \in L^2(\Omega)$ with $\boldsymbol{\beta} \cdot \nabla u \in L^2(\Omega)$.

We recall the discontinuous Galerkin discretization (20) to this problem and write it as follows: find $u_h \in V_h^p$ such that

$$a(u_h, v_h) = l(v_h) \quad \forall v_h \in V_h^p, \tag{22}$$

where $a(\cdot, \cdot)$ and $l(\cdot)$ are given by

$$\begin{aligned} a(w, v) = \sum_{\kappa \in \mathcal{T}_h} \Bigg\{ &\int_\kappa (\boldsymbol{\beta} \cdot \nabla w + bw) v \, \mathrm{d}\mathbf{x} \\ &- \int_{\partial\kappa_- \setminus \Gamma_-} \boldsymbol{\beta} \cdot \mathbf{n}\, [w] v^+ \, \mathrm{d}s - \int_{\partial\kappa_- \cap \Gamma_-} \boldsymbol{\beta} \cdot \mathbf{n}\, w^+ v^+ \, \mathrm{d}s \Bigg\}, \end{aligned} \tag{23}$$

with $[w] = w^+ - w^-$, and

$$l(w) = \sum_{\kappa \in \mathcal{T}_h} \left\{ \int_\kappa f v \, \mathrm{d}\mathbf{x} - \int_{\partial\kappa_- \cap \Gamma_-} \boldsymbol{\beta} \cdot \mathbf{n}\, g v^+ \, \mathrm{d}s \right\}.$$

Furthermore, we define the norms $\|v\|_\kappa \equiv \|v\|_{L^2(\kappa)} = \left(\int_\kappa |v|^2 \, d\mathbf{x}\right)^{\frac{1}{2}}$ for $\kappa \in \mathcal{T}_h$, and $\|v\|_e = \left(\int_e |\boldsymbol{\beta} \cdot \mathbf{n}||v|^2 \, d\mathbf{x}\right)^{\frac{1}{2}}$ for $e \subset \partial\kappa$. Then the following lemma can be obtained by performing integration by parts.

Lemma 2.1 *Let $a(\cdot,\cdot)$ be the bilinear form defined in* (23)*, then the following identity holds*

$$a(v,v) = \sum_{\kappa\in\mathcal{T}_h} \left\{ \int_\kappa \left(b - \tfrac{1}{2}\nabla\cdot\boldsymbol{\beta}\right) v^2 \, dx + \tfrac{1}{2}\|v^+\|^2_{\partial\kappa_-\cap\Gamma_-} \right. \\ \left. + \tfrac{1}{2}\|v^+ - v^-\|^2_{\partial\kappa_-\setminus\Gamma_-} + \tfrac{1}{2}\|v^+\|^2_{\partial\kappa_+\cap\Gamma_+} \right\}. \quad (24)$$

An implication of this Lemma is a stability result quoted in the following Lemma 2.2 that is a special case of Lemma 2.4 in [25] with $\delta = 0$, i.e. without streamline diffusion stabilization.

Lemma 2.2 (Stability) *Suppose that there exists a positive constant c_0 such that*

$$c(\mathbf{x}) := b(\mathbf{x}) - \tfrac{1}{2}\nabla\cdot\boldsymbol{\beta}(\mathbf{x}) \geq c_0, \quad x \in \bar{\Omega}. \quad (25)$$

Then the discrete solution $u_h \in V_h^p$ of (22) *obeys the bound*

$$\sum_{\kappa\in\mathcal{T}_h} \left\{ c_0\|u_h\|^2_\kappa + \|u_h^+ - u_h^-\|^2_{\partial\kappa_-\setminus\Gamma_-} + \|u_h^+\|^2_{\partial\kappa_+\cap\Gamma_+} + \tfrac{1}{2}\|u_h^+\|^2_{\partial\kappa_-\cap\Gamma_-} \right\} \\ \leq \tfrac{1}{c_0}\|f\|^2_\Omega + 2\|g\|^2_{\Gamma_-}.$$

Lemma 2.2 implies the uniqueness of the solution u_h of the discontinuous Galerkin method (22); furthermore, since (22) is a linear problem over the finite-dimensional space V_h^p, the existence of the solution u_h follows from its uniqueness.

Stimulated by the identity in Lemma (2.1) and definition (25), we define the DG-Norm $|||\cdot|||$ by

$$|||v||| := \sum_{\kappa\in\mathcal{T}_h} \left\{ \|cv\|^2_\kappa + \tfrac{1}{2}\|v^+\|^2_{\partial\kappa_-\cap\Gamma_-} + \tfrac{1}{2}\|v^+ - v^-\|^2_{\partial\kappa_-\setminus\Gamma_-} + \tfrac{1}{2}\|v^+\|^2_{\partial\kappa_+\cap\Gamma_+} \right\}.$$

In terms of this norm the following convergence result for the discontinuous Galerkin method applied to the scalar hyperbolic equation (19) can be shown, that is (in a simplified version) quoted from [25].

Theorem 2.3 (Convergence rate of the DG method) *Let $\mathcal{T}_h$ consist of shape-regular quadrilateral elements. For all $\kappa \in \mathcal{T}_h$ let $u|_\kappa \in H^{k_\kappa+1}(\kappa)$, $k_\kappa \geq 0$. Then, for the discrete solution $u_h \in V_h^p$ of the discontinuous Galerkin problem* (22) *and for $0 \leq s_\kappa \leq min(p, k_\kappa)$ the following estimate holds:*

$$|||u - u_h|||^2 \leq C \sum_{\kappa\in\mathcal{T}_h} h_\kappa^{2s_\kappa+1} |u|^2_{s_\kappa+1,\kappa}, \quad (26)$$

where C is independent of s_κ and h_κ.

This result was first proven by Johnson and Pitkaränta in [27] for general triangulations. However, this result indicates half an order of convergence in the L^2-norm less than expected from a trial space V_h^p of order $p+1$. Also numerical results show a full $O(h^{p+1})$ order of convergence on virtually all meshes, see also Section 3. In fact, Richter obtained in [32] the full order of convergence in the L^2-norm for some structured two-dimensional but non-Cartesian meshes. But on the other hand, Peterson confirmed in [30] by considering so–called *Peterson meshes*, that $O(h^{p+1/2})$ is actually the optimal order of convergence in the L^2-norm on general meshes.

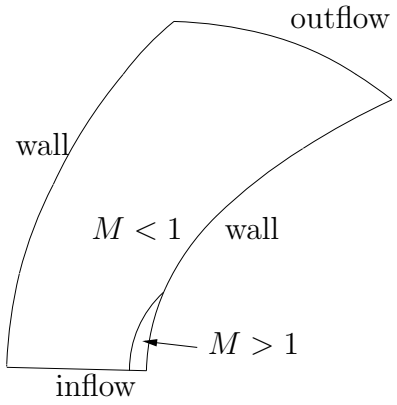


Figure 3: Geometry for Ringleb's flow; M denotes the Mach number.

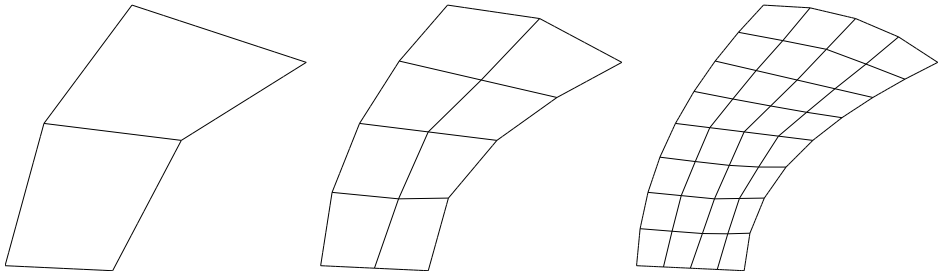


Figure 4: Ringleb flow problem: Meshes with 2, 8, 32 elements.

3 Higher order convergence of the DG method for the 2d compressible Euler equations

For nonlinear hyperbolic systems like the two–dimensional compressible Euler equations there are virtually no *a priori* error estimates available. Therefore, the order of convergence of these problems can be obtained through numerical tests, only. In the following, we shall show a numerical example taken from [19] indicating that the discontinuous Galerkin method applied to the steady 2d compressible Euler equation exhibits a full $O(h^{p+1})$ order of convergence for smooth solutions.

3.1 The Ringleb flow problem

To this end, we consider the solution of the 2d compressible Euler equations to the Ringleb flow problem, that is one of the few non-trivial problems of the 2d Euler equations for which a (smooth) analytical solution is known. For this case the analytical solution may be obtained using the hodograph transformation, see [14] or the appendix of [19]. This problem represents a transonic flow in a channel, see Figure 3, with inflow and outflow boundaries given by the lower and upper boundaries of the domain, and non–absorbing, i.e. *reflective* boundaries with normal velocity $\mathbf{v} \cdot \mathbf{n} = 0$, on the left and right boundary.

The solution to this flow problem is smooth but it is transonic with a small supersonic region near the lower right corner. Furthermore, this problem includes curved boundaries and the mesh consists of arbitrary quadrilaterals; see the first three globally refined meshes in Figure 4.

# el.	# DoFs	L^2-error	Rate	Order
2	32	8.48e-02	-	-
8	128	2.39e-02	3.54	1.82
32	512	6.72e-03	3.56	1.83
128	2048	1.81e-03	3.72	1.90
512	8192	4.75e-04	3.80	1.93
2048	32768	1.23e-04	3.87	1.95
8192	131072	3.14e-05	3.92	1.97

Table 1: Ringleb flow problem: DG(1) is of order $O(h^2)$.

# el.	# DoFs	L^2-error	Rate	Order
2	72	2.06e-02	-	-
8	288	2.75e-03	7.52	2.91
32	1152	3.85e-04	7.13	2.83
128	4608	5.18e-05	7.43	2.89
512	18432	6.61e-06	7.83	2.97
2048	73728	8.22e-07	8.05	3.01

Table 2: Ringleb flow problem: DG(2) is of order $O(h^3)$.

We note that low order boundary approximations of *reflective* boundaries reduce the order of convergence. To suppress this effect, here we impose the boundary condition,

$$B^-(\mathbf{u}, \mathbf{n})\,(\mathbf{u} - \mathbf{g}) = \mathbf{0}, \quad \text{on } \Gamma, \tag{27}$$

on the *whole* boundary Γ of the domain, where $\mathbf{g}$ is the boundary value function taken from the exact solution to the Ringleb flow problem. This boundary condition represents an inflow boundary condition for characteristic variables on inflow parts (with respect to the corresponding characteristics) of the boundary.

In the following, we compute the numerical solutions to this problem on globally refined meshes and evaluate the L^2-error of the solutions. The results of this test are presented in Tables 1- 4. They show the number of elements of the globally refined meshes, the number of degrees of freedom, the L^2-error of the numerical solution and two additional columns including the rate and the order of the convergence. These tables clearly show a $O(h^{p+1})$ order of convergence of the L^2-error $\|u - u_h\|$ of the numerical solution u_h. Finally Figure 5 shows the L^2-error of the DG(p), $0 \leq p \leq 4$ methods plotted against the number of degrees of freedom. The resulting $O(h^{p+1})$ order of convergence is optimal for trial and test functions of polynomial degree p. This again, as discussed in Section 2.6 for the scalar case, is half an order more than expected for the scalar linear case on general meshes. We re-emphasise that, although the discontinuous Galerkin method is based on ideas of upwinding, it is *not* restricted to convergence of first or second order but allows arbitrarily high orders of convergence (for smooth solutions) depending on the order of the discrete function space $\mathbf{V}_h^p$ employed.

# el.	# DoFs	L^2-error	Rate	Order
2	128	5.85e-03	-	-
8	512	3.58e-04	16.37	4.03
32	2048	2.84e-05	12.60	3.66
128	8192	1.92e-06	14.78	3.89
512	32768	1.23e-07	15.61	3.96
2048	131072	7.81e-09	15.75	3.98

Table 3: Ringleb flow problem: DG(3) is of order $O(h^4)$.

# el.	# DoFs	L^2-error	Rate	Order
2	200	2.92e-03	-	-
8	800	6.92e-05	42.13	5.40
32	3200	2.25e-06	30.81	4.95
128	12800	7.77e-08	28.94	4.86
512	51200	2.56e-09	30.32	4.92
2048	204800	8.21e-11	31.21	4.96

Table 4: Ringleb flow problem: DG(4) is of order $O(h^5)$.

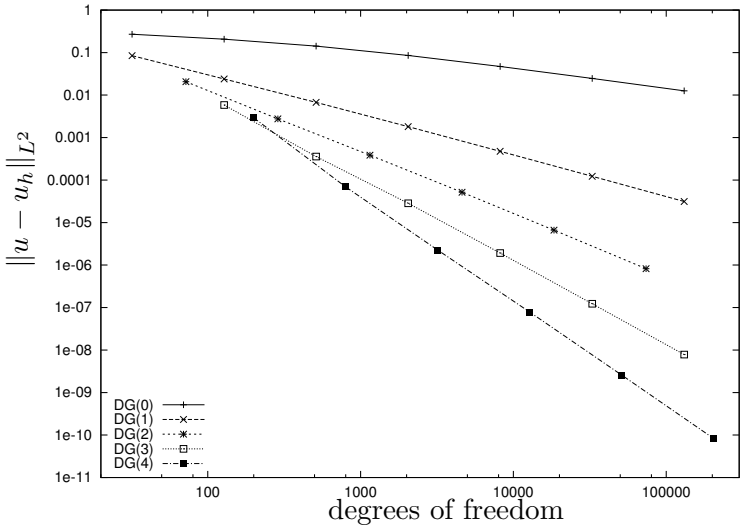


Figure 5: Ringleb flow problem: L^2-error of DG(p), $0 \leq p \leq 4$, solutions.

4 A Discontinuous Galerkin discretization of the compressible Navier-Stokes equations

4.1 The compressible Navier-Stokes equations

We consider the two–dimensional steady state compressible Navier-Stokes equations. Like in Section 2.3, ρ, $\mathbf{v} = (v_1, v_2)^T$, p and E denote the density, velocity vector, pressure and specific total energy, respectively. Furthermore, T denotes the temperature. The equations of motion are given by

$$\nabla \cdot (\mathcal{F}^c(\mathbf{u}) - \mathcal{F}^v(\mathbf{u}, \nabla\mathbf{u})) \equiv \frac{\partial}{\partial x_i}\mathbf{f}_i^c(\mathbf{u}) - \frac{\partial}{\partial x_i}\mathbf{f}_i^v(\mathbf{u}, \nabla\mathbf{u}) = 0 \quad \text{in } \Omega. \tag{28}$$

The vector of conservative variables $\mathbf{u}$ and the convective fluxes $\mathbf{f}_i^c$, $i = 1, 2$, are given by (8). Furthermore, the viscous fluxes $\mathbf{f}_i^v$, $i = 1, 2$, are defined by

$$\mathbf{f}_1^v(\mathbf{u}, \nabla\mathbf{u}) = \begin{bmatrix} 0 \\ \tau_{11} \\ \tau_{21} \\ \tau_{1j}v_j + \mathcal{K}T_{x_1} \end{bmatrix} \quad \text{and} \quad \mathbf{f}_2^v(\mathbf{u}, \nabla\mathbf{u}) = \begin{bmatrix} 0 \\ \tau_{12} \\ \tau_{22} \\ \tau_{2j}v_j + \mathcal{K}T_{x_2} \end{bmatrix}, \tag{29}$$

respectively, where $\mathcal{K}$ is the thermal conductivity coefficient. Finally, the viscous stress tensor is defined by

$$\tau = \mu\left(\nabla\mathbf{v} + (\nabla\mathbf{v})^T - \tfrac{2}{3}(\nabla\cdot\mathbf{v})I\right), \tag{30}$$

where μ is the dynamic viscosity coefficient, and the temperature T is given by $e = c_v T$; thus

$$\mathcal{K}T = \tfrac{\mu\gamma}{Pr}\left(E - \tfrac{1}{2}\mathbf{v}^2\right),$$

where $Pr = 0.72$ is the Prandtl number.

For the purposes of discretization, we rewrite the compressible Navier–Stokes equations (28) in the following (equivalent) form:

$$\frac{\partial}{\partial x_i}\left(\mathbf{f}_i^c(\mathbf{u}) - G_{ij}(\mathbf{u})\frac{\partial\mathbf{u}}{\partial x_j}\right) = 0 \quad \text{in } \Omega. \tag{31}$$

Here, the matrices $G_{ij}(\mathbf{u}) = \partial\mathbf{f}_i^v(\mathbf{u}, \nabla\mathbf{u})/\partial u_{x_j}$, for $i, j = 1, 2$, i.e., $\mathbf{f}_i^v(\mathbf{u}, \nabla\mathbf{u}) = G_{ij}(\mathbf{u})\partial\mathbf{u}/\partial x_j$, $i = 1, 2$, where

$$G_{11} = \frac{\mu}{\rho}\begin{pmatrix} 0 & 0 & 0 & 0 \\ -\frac{4}{3}v_1 & \frac{4}{3} & 0 & 0 \\ -v_2 & 0 & 1 & 0 \\ -\left(\frac{4}{3}v_1^2 + v_2^2 + \frac{\gamma}{Pr}\left(E - \mathbf{v}^2\right)\right) & \left(\frac{4}{3} - \frac{\gamma}{Pr}\right)v_1 & \left(1 - \frac{\gamma}{Pr}\right)v_2 & \frac{\gamma}{Pr} \end{pmatrix},$$

$$G_{12} = \frac{\mu}{\rho}\begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{2}{3}v_2 & 0 & -\frac{2}{3} & 0 \\ -v_1 & 1 & 0 & 0 \\ -\frac{1}{3}v_1v_2 & v_2 & -\frac{2}{3}v_1 & 0 \end{pmatrix}, G_{21} = \frac{\mu}{\rho}\begin{pmatrix} 0 & 0 & 0 & 0 \\ -v_2 & 0 & 1 & 0 \\ \frac{2}{3}v_1 & -\frac{2}{3} & 0 & 0 \\ -\frac{1}{3}v_1v_2 & -\frac{2}{3}v_2 & v_1 & 0 \end{pmatrix},$$

$$G_{22} = \frac{\mu}{\rho}\begin{pmatrix} 0 & 0 & 0 & 0 \\ -v_1 & 1 & 0 & 0 \\ -\frac{4}{3}v_2 & 0 & \frac{4}{3} & 0 \\ -\left(v_1^2 + \frac{4}{3}v_2^2 + \frac{\gamma}{Pr}\left(E - \mathbf{v}^2\right)\right) & \left(1 - \frac{\gamma}{Pr}\right)v_1 & \left(\frac{4}{3} - \frac{\gamma}{Pr}\right)v_2 & \frac{\gamma}{Pr} \end{pmatrix}.$$

Given that $\Omega \subset \mathbb{R}^2$ is a bounded region, with boundary Γ, the system of conservation equations (31) must be supplemented by appropriate boundary conditions. For simplicity of presentation, we assume that Γ may be decomposed as follows

$$\Gamma = \Gamma_{\text{D,sup}} \cup \Gamma_{\text{D,sub-in}} \cup \Gamma_{\text{D,sub-out}} \cup \Gamma_{\text{N}} \cup \Gamma_{\text{W}},$$

where $\Gamma_{\text{D,sup}}$, $\Gamma_{\text{D,sub-in}}$, $\Gamma_{\text{D,sub-out}}$, Γ_{N}, and Γ_{W} are distinct subsets of Γ representing Dirichlet (supersonic), Dirichlet (subsonic-inflow), Dirichlet (subsonic-outflow), Neumann (supersonic-outflow), cf. the four cases mentioned in Section 2.3, and solid wall boundaries, respectively. Thereby, we may specify the following boundary conditions:

$$\mathcal{B}(\mathbf{u}) = \mathcal{B}(\mathbf{g}_D) \quad \text{on} \quad \Gamma_{\text{D,sup}} \cup \Gamma_{\text{D,sub-in}} \cup \Gamma_{\text{D,sub-out}}, \tag{32}$$

where $\mathbf{g}_{\text{D}}$ are given Dirichlet boundary conditions, respectively. Here, $\mathcal{B}$ is a boundary operator employed to enforce appropriate Dirichlet conditions on $\Gamma_{\text{D,sup}} \cup \Gamma_{\text{D,sub-in}} \cup \Gamma_{\text{D,sub-out}}$. For simplicity of presentation, we assume that

$$\mathcal{B}(\mathbf{u}) = \begin{cases} \mathbf{u}, & \text{on } \Gamma_{\text{D,sup}}, \\ (u_1, u_2, u_3, 0)^T, & \text{on } \Gamma_{\text{D,sub-in}}, \text{and} \\ (0, 0, 0, (\gamma - 1)(u_4 - (u_2^2 + u_3^2)/(2u_1)))^T, & \text{on } \Gamma_{\text{D,sub-out}} \end{cases} \tag{33}$$

We note that the latter condition enforces a specific pressure $p_{\text{out}} = (\mathcal{B}(\mathbf{g}_D))_4$ on $\Gamma_{\text{D,sub-out}}$. Furthermore, we specify Neumann boundary conditions

$$\mathcal{F}^v(\mathbf{u}, \nabla \mathbf{u}) \cdot \mathbf{n} = \mathbf{g}_{\text{N}} \quad \text{on} \quad \Gamma_{\text{N}}, \tag{34}$$

with $\mathbf{g}_N = 0$ on the supersonic outflow boundary Γ_{N}.

For solid wall boundaries and *viscous* flows, we consider the distinction between *isothermal* and *adiabatic* conditions. To this end, decomposing $\Gamma_W = \Gamma_{\text{W,iso}} \cup \Gamma_{\text{W,adia}}$, we set

$$\mathbf{v} = \mathbf{0} \quad \text{on} \quad \Gamma_{\text{W}}, \qquad T = T_{\text{wall}} \quad \text{on} \quad \Gamma_{\text{W,iso}}, \qquad \mathbf{n} \cdot \nabla T = 0 \quad \text{on} \quad \Gamma_{\text{W,adia}},$$

where T_{wall} is a given wall temperature;

For solid wall boundaries and *inviscid* flows, governed by the compressible Euler equations, see Section 2.3 or equation (28) with $\mu = 0$, we consider *reflective* (or slip wall) boundary conditions,

$$\mathbf{v} \cdot \mathbf{n} = 0 \quad \text{on} \quad \Gamma_{\text{W}} = \Gamma_{\text{refl}}.$$

4.2 The Interior Penalty Discontinuous Galerkin discretization

Following the derivation in [23], in this section we introduce the discontinuous Galerkin method with interior penalty for the discretization of the compressible Navier–Stokes equations (28).

In addition to the notation introduced in Section 2.4, we now define average and jump operators. To this end, let κ^+ and κ^- be two adjacent elements of $\mathcal{T}_h$ and $\mathbf{x}$ be an arbitrary point on the interior edge $e = \partial\kappa^+ \cap \partial\kappa^- \subset \Gamma_{\mathcal{I}}$, where $\Gamma_{\mathcal{I}}$ denotes the union of all interior edges of $\mathcal{T}_h$. Moreover, let $\mathbf{v}$ and $\underline{\tau}$ be vector- and matrix-valued functions, respectively, that are smooth inside each element $\kappa^\pm$. By $(\mathbf{v}^\pm, \underline{\tau}^\pm)$ we denote the traces of $(\mathbf{v}, \underline{\tau})$ on e taken from within the interior of $\kappa^\pm$, respectively. Then, we define the averages at $\mathbf{x} \in e$ by $\{\!\{\mathbf{v}\}\!\} = (\mathbf{v}^+ + \mathbf{v}^-)/2$ and $\{\!\{\underline{\tau}\}\!\} = (\underline{\tau}^+ + \underline{\tau}^-)/2$. Similarly, the jump at $\mathbf{x} \in e$ is given by $\underline{[\![\mathbf{v}]\!]} = \mathbf{v}^+ \otimes \mathbf{n}_{\kappa^+} + \mathbf{v}^- \otimes \mathbf{n}_{\kappa^-}$. On a boundary edge $e \subset \Gamma$, we set

$\{\!\{\mathbf{v}\}\!\} = \mathbf{v}$, $\{\!\{\underline{\tau}\}\!\} = \underline{\tau}$ and $[\![\underline{\mathbf{v}}]\!] = \mathbf{v} \otimes \mathbf{n}$. For matrices $\underline{\sigma}, \underline{\tau} \in \mathbb{R}^{m\times n}$, $m, n \geq 1$, we use the standard notation $\underline{\sigma} : \underline{\tau} = \sum_{k=1}^{m} \sum_{l=1}^{n} \sigma_{kl}\tau_{kl}$; additionally, for vectors $\mathbf{v} \in \mathbb{R}^m, \mathbf{w} \in \mathbb{R}^n$, the matrix $\mathbf{v} \otimes \mathbf{w} \in \mathbb{R}^{m\times n}$ is defined by $(\mathbf{v} \otimes \mathbf{w})_{kl} = v_k\, w_l$.

In order to derive the interior penalty discontinuous Galerkin discretization of equations (31), we first re-write (31) as a system of first–order partial differential equations, by introducing appropriate auxiliary variables. To this end, we have

$$\underline{\sigma} = \left(G_{1j}(\mathbf{u}) \frac{\partial \mathbf{u}}{\partial x_j}, G_{2j}(\mathbf{u}) \frac{\partial \mathbf{u}}{\partial x_j} \right), \quad \text{in } \Omega, \tag{35}$$

$$\nabla \cdot (\mathcal{F}^c(\mathbf{u}) - \underline{\sigma}) = 0, \quad \text{in } \Omega, \tag{36}$$

subject to appropriate boundary conditions, cf. above.

Next, we introduce the finite element space $\underline{\Sigma}_h^p \times \mathbf{V}_h^p$, where

$$\begin{aligned} \underline{\Sigma}_h^p &= \{\underline{\tau} \in [L_2(\Omega)]^{4\times 2} : \underline{\tau}|_\kappa \circ \sigma_\kappa \in [\mathcal{Q}_p(\hat{\kappa})]^{4\times 2},\ \kappa \in \mathcal{T}_h\}, \\ \mathbf{V}_h^p &= \{\mathbf{v} \in [L_2(\Omega)]^4 : \mathbf{v}|_\kappa \circ \sigma_\kappa \in [\mathcal{Q}_p(\hat{\kappa})]^4,\ \kappa \in \mathcal{T}_h\}, \end{aligned} \tag{37}$$

for an approximation order $p \geq 1$. Here, $\mathbf{V}_h^p$ is defined as in (13) for $m = 4$ and $\mathcal{Q}_p(\hat{\kappa})$ denotes the space of tensor product polynomials on $\hat{\kappa}$ of degree p in each coordinate direction as defined in (12).

Taking the $L_2(\kappa)$, $\kappa \in \mathcal{T}_h$, inner product of (35) and (36) with smooth test functions $\underline{\tau} = (\underline{\tau}_1, \underline{\tau}_2)$ and $\mathbf{v}$, respectively, and integrating by parts gives

$$\begin{aligned} \int_\kappa \underline{\sigma} : \underline{\tau}\, \mathrm{d}\mathbf{x} &= \int_{\partial\kappa} \mathbf{u} \cdot (G_{ij}^T \underline{\tau}_i) n_{x_j}\, \mathrm{d}s - \int_\kappa \mathbf{u} \cdot \frac{\partial}{\partial x_j}(G_{ij}^T \underline{\tau}_i)\, \mathrm{d}\mathbf{x}, \\ - \int_\kappa \mathcal{F}^c(\mathbf{u}) : \nabla \mathbf{v}\, \mathrm{d}\mathbf{x} + \int_{\partial\kappa} (\mathcal{F}^c(\mathbf{u}) \cdot \mathbf{n}_\kappa) \cdot \mathbf{v}\, \mathrm{d}s + \int_\kappa \underline{\sigma} : \nabla \mathbf{v}\, \mathrm{d}\mathbf{x} \\ - \int_{\partial\kappa \setminus (\Gamma_{\mathrm{N}} \cup \Gamma_{\mathrm{W,adia}})} \underline{\sigma} : \mathbf{v} \otimes \mathbf{n}_\kappa\, \mathrm{d}s - \int_{\partial\kappa \cap \Gamma_{\mathrm{N}}} \mathbf{g}_{\mathrm{N}} \cdot \mathbf{v}\, \mathrm{d}s \\ - \int_{\partial\kappa \cap \Gamma_{\mathrm{W,adia}}} \underline{\sigma}^{\mathrm{adia}} : \mathbf{v} \otimes \mathbf{n}_\kappa\, \mathrm{d}s &= 0, \end{aligned}$$

where on the adiabatic boundary $\Gamma_{\mathrm{W,adia}}$, we define $\underline{\sigma}^{\mathrm{adia}}$ such that

$$\underline{\sigma}^{\mathrm{adia}} \cdot \mathbf{n} \equiv \mathcal{F}^{v,\mathrm{adia}}(\mathbf{u}, \nabla \mathbf{u}) \cdot \mathbf{n} = (0, \tau_{1j} n_{x_j}, \tau_{2j} n_{x_j}, \tau_{ij} v_j n_{x_i})^T.$$

Summing over all elements κ in the computational mesh $\mathcal{T}_h$ and introducing appropriate numerical flux functions, which will be defined below, we deduce the following auxiliary formulation of the interior penalty discontinuous Galerkin discretization: find

$(\underline{\sigma}_h, \mathbf{u}_h) \in \underline{\Sigma}_h^p \times \mathbf{V}_h^p$ such that

$$\sum_{\kappa\in\mathcal{T}_h}\int_\kappa \underline{\sigma}_h : \underline{\tau}_h \,\mathrm{d}\mathbf{x} = \sum_{\kappa\in\mathcal{T}_h}\left\{\int_{\partial\kappa} \widehat{\mathbf{u}}_h\cdot(G_{ij}^T\underline{\tau}_{h,i})n_{x_j}\,\mathrm{d}s - \int_\kappa \mathbf{u}_h\cdot\frac{\partial}{\partial x_j}(G_{ij}^T\underline{\tau}_{h,i})\,\mathrm{d}\mathbf{x}\right\}, \tag{38}$$

$$\sum_{\kappa\in\mathcal{T}_h}\left\{-\int_\kappa \mathcal{F}^c(\mathbf{u}_h):\nabla\mathbf{v}_h\,\mathrm{d}\mathbf{x} + \int_{\partial\kappa\setminus\Gamma}\mathcal{H}(\mathbf{u}_h^+,\mathbf{u}_h^-,\mathbf{n}_\kappa)\cdot\mathbf{v}_h^+\,\mathrm{d}s + \int_{\partial\kappa\cap\Gamma}\mathcal{H}(\mathbf{u}_h^+,\mathbf{u}_\Gamma(\mathbf{u}_h^+),\mathbf{n}_\kappa)\cdot\mathbf{v}_h^+\,\mathrm{d}s + \int_\kappa \underline{\sigma}_h:\nabla\mathbf{v}_h\,\mathrm{d}\mathbf{x}\right\} \tag{39}$$

$$-\int_{\Gamma_{\mathcal{I}}\cup\Gamma\setminus(\Gamma_{\mathrm{N}}\cup\Gamma_{\mathrm{W,adia}})}\widehat{\underline{\sigma}}_h:\underline{[\![\mathbf{v}_h]\!]}\,\mathrm{d}s - \int_{\Gamma_{\mathrm{W,adia}}}\widehat{\underline{\sigma}}_h^{\mathrm{adia}}:\underline{[\![\mathbf{v}_h]\!]}\,\mathrm{d}s - \int_{\Gamma_{\mathrm{N}}}\mathbf{g}_{\mathrm{N}}\cdot\mathbf{v}_h\,\mathrm{d}s = 0$$

for all $(\underline{\tau}_h, \mathbf{v}_h) \in \underline{\Sigma}_h^p \times \mathbf{V}_h^p$. Here, the numerical flux function $\mathcal{H}(\cdot,\cdot,\cdot)$ is defined as in Section 2.4. Furthermore, the numerical flux functions $\widehat{\mathbf{u}}_h$, $\widehat{\underline{\sigma}}_h$ and $\widehat{\underline{\sigma}}_h^{\mathrm{adia}}$ may be chosen to be any Lipschitz continuous, consistent and conservative fluxes which are discrete approximations to traces on the boundary of the elements in the mesh.

To define the symmetric interior penalty discretization of the viscous terms, these numerical fluxes are defined as follows: for an edge e which lies inside the domain Ω, we have

$$\widehat{\mathbf{u}}_h = \{\!\{\mathbf{u}_h\}\!\} \qquad \text{and} \qquad \widehat{\underline{\sigma}}_h = \{\!\{\mathcal{F}^v(\mathbf{u}_h,\nabla\mathbf{u}_h)\}\!\} - \delta\underline{[\![\mathbf{u}_h]\!]};$$

while for boundary edges we write

$$\widehat{\mathbf{u}}_h = \mathbf{u}_\Gamma(\mathbf{u}_h^+), \qquad \widehat{\underline{\sigma}}_h = \mathcal{F}^v(\mathbf{u}_h^+,\nabla\mathbf{u}_h^+) - \delta(\mathbf{u}_h^+ - \mathbf{u}_\Gamma(\mathbf{u}_h^+))\otimes\mathbf{n},$$

and on $\Gamma_{\mathrm{W,adia}}$, we set

$$\widehat{\underline{\sigma}}_h^{\mathrm{adia}} = \mathcal{F}^{v,\mathrm{adia}}(\mathbf{u}_h^+,\nabla\mathbf{u}_h^+) - \delta(\mathbf{u}_h^+ - \mathbf{u}_\Gamma(\mathbf{u}_h^+))\otimes\mathbf{n}. \tag{40}$$

Here, for simplicity, we set the discontinuity penalization matrix $\delta = \mathrm{diag}\{\delta_i, i = 1,\dots,4\}$, where

$$\delta_i|_e = C_{\mathrm{IP}}\frac{\mu p^2}{\tilde{h}} \quad \text{for } e\subset\Gamma_{\mathcal{I}}\cup\Gamma, \tag{41}$$

$\tilde{h} = \min(\mathrm{meas}(\kappa),\mathrm{meas}(\kappa'))/\mathrm{meas}(e)$ represents the element dimension orthogonal to the edge e of elements κ and κ' adjacent to e, and C_{IP} is a positive constant, which, for reasons of stability, must be chosen sufficiently large, cf. [2]; see also [17] for the extension to the anisotropic case.

Finally, the boundary function $\mathbf{u}_\Gamma(\mathbf{u})$ is given according to the type of boundary condition imposed. To this end, we set

$$\mathbf{u}_\Gamma(\mathbf{u}) = \mathbf{g}_D \ \text{ on } \ \Gamma_{\mathrm{D,sup}}, \qquad \mathbf{u}_\Gamma(\mathbf{u}) = \mathbf{u} \ \text{ on } \ \Gamma_{\mathrm{N}},$$

$$\mathbf{u}_\Gamma(\mathbf{u}) = \left((g_D)_1,(g_D)_2,(g_D)_3,\frac{p(\mathbf{u})}{\gamma-1}+\frac{(g_D)_2^2+(g_D)_3^2}{2(g_D)_1}\right)^T \ \text{ on } \ \Gamma_{\mathrm{D,sub\text{-}in}}, \tag{42}$$

and

$$\mathbf{u}_\Gamma(\mathbf{u}) = \left(u_1,u_2,u_3,\frac{p_{\mathrm{out}}}{\gamma-1}+\frac{u_2^2+u_3^2}{2u_1}\right)^T \ \text{ on } \ \Gamma_{\mathrm{D,sub\text{-}out}}. \tag{43}$$

Here, $p \equiv p(\mathbf{u})$ denotes the pressure evaluated using the equation of state (10). For viscous flows, we set

$$\mathbf{u}_\Gamma(\mathbf{u}) = (u_1, 0, 0, u_1 c_v T_{\text{wall}})^T \quad \text{on} \quad \Gamma_{\text{W,iso}},$$

and

$$\mathbf{u}_\Gamma(\mathbf{u}) = (u_1, 0, 0, u_4)^T \quad \text{on} \quad \Gamma_{\text{W,adia}}.$$

Finally, for inviscid flows, we set

$$\mathbf{u}_\Gamma(\mathbf{u}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1-2n_1^2 & -2n_1n_2 & 0 \\ 0 & -2n_1n_2 & 1-2n_2^2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{u} \quad \text{on} \quad \Gamma_{\text{refl}},$$

which originates from $\mathbf{u}$ by inverting the sign of the normal velocity component of $\mathbf{u}$, i.e. $\mathbf{v} = (v_1, v_2)$ is replaced by $\mathbf{v}^- = (\mathbf{v} - 2(\mathbf{v}\cdot\mathbf{n})\mathbf{n})$.

Remark. We note that the flux functions $\widehat{\underline{\sigma}}_h$ and $\widehat{\underline{\sigma}}_h^{\text{adia}}$ are consistent for any choice of δ; however, it is well known that the stability of the underlying discretization crucially depends on the magnitude of this discontinuity–penalization parameter, cf. [2], for example. For the choice of the corresponding numerical flux functions for the non-symmetric interior penalty method, together with other schemes proposed in the literature, we refer to the article [2].

It is usually desirable to eliminate the auxiliary variables $\underline{\sigma}_h$, in order to reduce the size of the underlying system of nonlinear equations. This can be achieved by selecting $\underline{\tau}_h = \nabla \mathbf{v}_h$ in (38), integrating by parts, and inserting the resulting expression for the term involving the product of $\underline{\sigma}_h$ and $\nabla \mathbf{v}_h$ into (39). Thereby, the so–called primal formulation of the symmetric interior penalty discontinuous Galerkin discretization of the compressible Navier–Stokes equations (31) is as follows: find $\mathbf{u}_h \in \mathbf{V}_h^p$ such that

$$\begin{aligned}
\mathcal{N}(\mathbf{u}_h, \mathbf{v}_h) \equiv & -\int_\Omega \mathcal{F}^c(\mathbf{u}_h) : \nabla_h \mathbf{v}_h \,\mathrm{d}\mathbf{x} + \sum_{\kappa\in\mathcal{T}_h} \int_{\partial\kappa\setminus\Gamma} \mathcal{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n}_\kappa)\cdot \mathbf{v}_h^+ \,\mathrm{d}s \\
& + \int_\Omega \mathcal{F}^v(\mathbf{u}_h, \nabla_h \mathbf{u}_h) : \nabla_h \mathbf{v}_h \,\mathrm{d}\mathbf{x} - \int_{\Gamma_\mathcal{I}} \{\!\!\{ \mathcal{F}^v(\mathbf{u}_h, \nabla_h \mathbf{u}_h) \}\!\!\} : \underline{[\![\mathbf{v}_h]\!]} \,\mathrm{d}s \\
& - \int_{\Gamma_\mathcal{I}} \{\!\!\{ \left(G_{i1}^T \partial_h \mathbf{v}_h / \partial x_i, G_{i2}^T \partial_h \mathbf{v}_h / \partial x_i\right) \}\!\!\} : \underline{[\![\mathbf{u}_h]\!]} \,\mathrm{d}s + \int_{\Gamma_\mathcal{I}} \delta \underline{[\![\mathbf{u}_h]\!]} : \underline{[\![\mathbf{v}_h]\!]} \,\mathrm{d}s \\
& + \int_\Gamma \mathcal{H}(\mathbf{u}_h^+, \mathbf{u}_\Gamma(\mathbf{u}_h^+), \mathbf{n}) \cdot \mathbf{v}_h^+ \,\mathrm{d}s + \int_{\Gamma\setminus\Gamma_N} \delta \left(\mathbf{u}_h^+ - \mathbf{u}_\Gamma(\mathbf{u}_h^+)\right) \cdot \mathbf{v}_h^+ \,\mathrm{d}s, \\
& - \int_{\Gamma\setminus(\Gamma_N \cup \Gamma_{\text{W,adia}})} \mathcal{F}^v(\mathbf{u}_h^+, \nabla_h \mathbf{u}_h^+) : \underline{[\![\mathbf{v}_h]\!]} \,\mathrm{d}s - \int_{\Gamma_{\text{N}}} \mathbf{g}_{\text{N}} \cdot \mathbf{v}_h \,\mathrm{d}s \\
& - \int_{\Gamma_{\text{W,adia}}} \mathcal{F}^{v,\text{adia}}(\mathbf{u}_h^+, \nabla_h \mathbf{u}_h^+) : \underline{[\![\mathbf{v}_h]\!]} \,\mathrm{d}s \\
& - \int_{\Gamma\setminus\Gamma_{\text{N}}} \left(G_{i1}^T(\mathbf{u}_h^+) \partial_h \mathbf{v}_h^+ / \partial x_i, G_{i2}^T(\mathbf{u}_h^+) \partial_h \mathbf{v}_h^+ / \partial x_i\right) : \left(\mathbf{u}_h^+ - \mathbf{u}_\Gamma(\mathbf{u}_h^+)\right) \otimes \mathbf{n} \,\mathrm{d}s = 0 \qquad (44)
\end{aligned}$$

for all $\mathbf{v}_h$ in $\mathbf{V}_h^p$. Here, the subscript h on the operators ∇_h and $\partial_h/\partial x_i$, $i = 1, 2$, is used to denote the discrete counterparts of ∇ and $\partial/\partial x_i$, $i = 1, 2$, respectively, taken elementwise.

5 Newton–GMRES algorithm

To determine the numerical solution $\mathbf{u}_h$ of the system of nonlinear equations (44), we employ a damped Newton method. This nonlinear iteration generates a sequence of approximations $\mathbf{u}_h^n$, $n = 0, 1, \ldots$, to the actual numerical solution $\mathbf{u}_h$, using the following algorithm. Given an iterate $\mathbf{u}_h^n$, the update $\mathbf{d}_h^n$ of $\mathbf{u}_h^n$ to get to the next iterate

$$\mathbf{u}_h^{n+1} = \mathbf{u}_h^n + \omega^n \mathbf{d}_h^n$$

is defined by: find $\mathbf{d}_h^n \in \mathbf{V}_h^p$ such that

$$\hat{\mathcal{N}}'_{\mathbf{u}}[\mathbf{u}_h^n](\mathbf{d}_h^n, \mathbf{v}_h) = R(\mathbf{u}_h^n, \mathbf{v}_h) \equiv -\mathcal{N}(\mathbf{u}_h^n, \mathbf{v}_h) \quad \forall \mathbf{v}_h \in \mathbf{V}_h^p. \tag{45}$$

Here, w^n denotes a damping parameter, which is dynamically chosen to guarantee that the discrete l_2-norm of the residual computed with $\mathbf{u}_h^{n+1}$ is less than the same quantity computed with $\mathbf{u}_h^n$; $w^n = 1$ represents an undamped Newton iteration. Additionally, $\hat{\mathcal{N}}'_{\mathbf{u}}[\mathbf{w}](\cdot, \mathbf{v})$ denotes (an approximation to) the Fréchet derivative of $\mathbf{u} \to \mathcal{N}(\mathbf{u}, \mathbf{v})$, for $\mathbf{v} \in \mathbf{V}_h^p$ fixed, at some $\mathbf{w}$ in $\mathbf{V}$, where $\mathbf{V}$ is some suitable chosen function space such that $\mathbf{V}_h^p \in \mathbf{V}$. $\hat{\mathcal{N}}'_{\mathbf{u}}[\mathbf{w}](\cdot, \mathbf{v})$ is also referred to as the *Jacobian* of the numerical scheme. More precisely, we approximate $\mathcal{N}'_{\mathbf{u}}[\mathbf{w}](\cdot, \mathbf{v})$ by

$$\begin{aligned}
\hat{\mathcal{N}}'_{\mathbf{u}}[\mathbf{w}](\boldsymbol{\phi}, \mathbf{v}) &= -\int_\Omega (\mathcal{F}^c_{\mathbf{u}}(\mathbf{w})\boldsymbol{\phi}) : \nabla_h \mathbf{v} \, \mathrm{d}\mathbf{x} \\
&+ \sum_{\kappa \in \mathcal{T}_h} \int_{\partial\kappa \setminus \Gamma} \left(\hat{\mathcal{H}}'_{\mathbf{u}^+}(\mathbf{w}^+, \mathbf{w}^-, \mathbf{n}_\kappa)\boldsymbol{\phi}^+ + \hat{\mathcal{H}}'_{\mathbf{u}^-}(\mathbf{w}^+, \mathbf{w}^-, \mathbf{n}_\kappa)\boldsymbol{\phi}^- \right) \cdot \mathbf{v}^+ \, \mathrm{d}s \\
&+ \int_\Omega (\mathcal{F}^v_{\mathbf{u}}(\mathbf{w}, \nabla_h \mathbf{w})\boldsymbol{\phi}) : \nabla_h \mathbf{v} \, \mathrm{d}\mathbf{x} + \int_\Omega (\mathcal{F}^v_{\nabla \mathbf{u}}(\mathbf{w}, \nabla_h \mathbf{w}) \nabla_h \boldsymbol{\phi}) : \nabla_h \mathbf{v} \, \mathrm{d}\mathbf{x} \\
&- \int_{\Gamma_\mathcal{I}} \{\!\{ \mathcal{F}^v_{\mathbf{u}}(\mathbf{w}, \nabla_h \mathbf{w})\boldsymbol{\phi} \}\!\} : \underline{[\![\mathbf{v}]\!]} \, \mathrm{d}s - \int_{\Gamma_\mathcal{I}} \{\!\{ \mathcal{F}^v_{\nabla \mathbf{u}}(\mathbf{w}, \nabla_h \mathbf{w}) \nabla_h \boldsymbol{\phi} \}\!\} : \underline{[\![\mathbf{v}]\!]} \, \mathrm{d}s \\
&- \int_{\Gamma_\mathcal{I}} \{\!\{ \left(\left(G^T_{i1} \right)' (\mathbf{w})\boldsymbol{\phi} \right) \partial_h \mathbf{v} / \partial x_i, \left(\left(G^T_{i2} \right)' (\mathbf{w})\boldsymbol{\phi} \right) \partial_h \mathbf{v} / \partial x_i \}\!\} : \underline{[\![\mathbf{w}]\!]}) \, \mathrm{d}s \\
&- \int_{\Gamma_\mathcal{I}} \{\!\{ \left(G^T_{i1}(\mathbf{w}) \partial_h \mathbf{v} / \partial x_i, G^T_{i2}(\mathbf{w}) \partial_h \mathbf{v} / \partial x_i \right) \}\!\} : \underline{[\![\boldsymbol{\phi}]\!]}) \, \mathrm{d}s + \int_{\Gamma_\mathcal{I}} \delta \underline{[\![\boldsymbol{\phi}]\!]} : \underline{[\![\mathbf{v}]\!]} \, \mathrm{d}s \\
&+ \hat{\mathcal{N}}'_{\Gamma, \mathbf{u}}[\mathbf{w}](\boldsymbol{\phi}, \mathbf{v}),
\end{aligned}$$

where $\mathbf{w} \to \hat{\mathcal{H}}'_{\mathbf{u}^+}(\mathbf{w}^+, \mathbf{w}^-, \mathbf{n}_\kappa)$ and $\mathbf{w} \to \hat{\mathcal{H}}'_{\mathbf{u}^-}(\mathbf{w}^+, \mathbf{w}^-, \mathbf{n}_\kappa)$ denote approximations to the derivatives of the flux function $\mathcal{H}(\cdot, \cdot, \cdot)$ with respect to its first and second arguments, respectively; for a detailed description of these derivatives for two specific choices of numerical fluxes, we refer to Subsection 5.2. A detailed description of the Jacobian $\hat{\mathcal{N}}'_{\Gamma, \mathbf{u}}[\mathbf{w}](\cdot, \mathbf{v})$ of the boundary terms is given in the following subsection.

5.1 Jacobians of boundary terms

$\hat{\mathcal{N}}'_{\Gamma,\mathbf{u}}[\mathbf{w}](\cdot,\mathbf{v})$ denotes the derivative of the boundary terms, which is given by

$$
\begin{aligned}
&\hat{\mathcal{N}}'_{\Gamma,\mathbf{u}}[\mathbf{w}](\boldsymbol{\phi},\mathbf{v}) = \\
&\int_\Gamma \left(\hat{\mathcal{H}}'_{\mathbf{u}^+}\left(\mathbf{w}^+,\mathbf{u}_\Gamma(\mathbf{w}^+),\mathbf{n}\right) + \hat{\mathcal{H}}'_{\mathbf{u}^-}\left(\mathbf{w}^+,\mathbf{u}_\Gamma(\mathbf{w}^+),\mathbf{n}\right)\mathbf{u}'_\Gamma(\mathbf{w}^+)\right)\boldsymbol{\phi}^+\cdot\mathbf{v}^+\,\mathrm{d}s \\
&+\int_{\Gamma\setminus\Gamma_N}\delta\left(\boldsymbol{\phi}^+ - \mathbf{u}'_\Gamma(\mathbf{w}^+)\boldsymbol{\phi}^+\right)\cdot\mathbf{v}^+\,\mathrm{d}s, \\
&-\int_{\Gamma\setminus(\Gamma_N\cup\Gamma_{\mathrm{W,adia}})}\left(\mathcal{F}^v_{\mathbf{u}}(\mathbf{w}^+,\nabla_h\mathbf{w}^+)\boldsymbol{\phi}^+ + \mathcal{F}^v_{\nabla\mathbf{u}}(\mathbf{w}^+,\nabla_h\mathbf{w}^+)\nabla_h\boldsymbol{\phi}^+\right):\underline{[\![\mathbf{v}]\!]}\,\mathrm{d}s \\
&-\int_{\Gamma_{\mathrm{W,adia}}}\left(\mathcal{F}^{v,\mathrm{adia}}_{\mathbf{u}}(\mathbf{w}^+,\nabla_h\mathbf{w}^+)\boldsymbol{\phi}^+ + \mathcal{F}^{v,\mathrm{adia}}_{\nabla\mathbf{u}}(\mathbf{w}^+,\nabla_h\mathbf{w}^+)\nabla_h\boldsymbol{\phi}^+\right):\underline{[\![\mathbf{v}]\!]}\,\mathrm{d}s \\
&-\int_{\Gamma\setminus\Gamma_{\mathrm{N}}}\Big(\Big(\left(G^T_{i1}\right)'(\mathbf{w}^+)\boldsymbol{\phi}^+\Big)\partial_h\mathbf{v}^+/\partial x_i,\Big(\left(G^T_{i2}\right)'(\mathbf{w}^+)\boldsymbol{\phi}^+\Big)\partial_h\mathbf{v}^+/\partial x_i\Big) \\
&\qquad:\left(\mathbf{w}^+ - \mathbf{u}_\Gamma(\mathbf{w}^+)\right)\otimes\mathbf{n}\,\mathrm{d}s \\
&-\int_{\Gamma\setminus\Gamma_{\mathrm{N}}}\left(G^T_{i1}(\mathbf{w}^+)\partial_h\mathbf{v}^+/\partial x_i, G^T_{i2}(\mathbf{w}^+)\partial_h\mathbf{v}^+/\partial x_i\right):\left(\boldsymbol{\phi}^+ - \mathbf{u}'_\Gamma(\mathbf{w}^+)\boldsymbol{\phi}^+\right)\otimes\mathbf{n}\,\mathrm{d}s,
\end{aligned}
$$

where $\mathbf{u}'_\Gamma(\mathbf{u})$ denotes the derivative of the boundary function $\mathbf{u}_\Gamma(\mathbf{u})$ with respect to the conservative variables (i.e. the components of) $\mathbf{u}$. On the supersonic parts of the boundary, we have

$$
\mathbf{u}'_\Gamma(\mathbf{u}) = 0 \ \text{ on } \ \Gamma_{\mathrm{D,sup}} \quad \text{and} \quad \mathbf{u}'_\Gamma(\mathbf{u}) = I \in \mathbb{R}^{4,4} \ \text{ on } \ \Gamma_{\mathrm{N}};
$$

on the subsonic boundaries $\Gamma_{\mathrm{D,sub\text{-}in}}$ and $\Gamma_{\mathrm{D,sub\text{-}out}}$, $\mathbf{u}'_\Gamma(\mathbf{u})$ is given by

$$
\mathbf{u}'_\Gamma(\mathbf{u}) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{2}|\mathbf{v}|^2 & -v_1 & -v_2 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{u}'_\Gamma(\mathbf{u}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\frac{1}{2}|\mathbf{v}|^2 & v_1 & v_2 & 0 \end{pmatrix},
$$

respectively. For viscous flows, we have

$$
\mathbf{u}'_\Gamma(\mathbf{u}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ c_v T_{\mathrm{wall}} & 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \mathbf{u}'_\Gamma(\mathbf{u}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}
$$

on the isothermal and adiabatic no-slip boundaries, respectively. Finally, for inviscid flows on the reflective (slip wall) boundaries, we have

$$
\mathbf{u}'_\Gamma(\mathbf{u}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1-2n_1^2 & -2n_1n_2 & 0 \\ 0 & -2n_1n_2 & 1-2n_2^2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.
$$

5.2 Jacobians of numerical flux functions

It remains to give an expression for the derivatives $\mathcal{H}'_{\mathbf{u}^+}$ and $\mathcal{H}'_{\mathbf{u}^-}$ of the numerical flux function $\mathcal{H}$. Clearly, they depend on the specific choice of the flux function and do – strictly speaking – not exist for many flux functions, as they typically include some non-differentiable terms such as 'min' and 'max' or absolute value functions, for example. Nevertheless, they can be approximated in practice, and, provided the approximation is sufficiently good, the resulting Jacobian is still capable of delivering an optimal convergence of the Newton iteration as demonstrated in Section 5.3.

First, we consider the local Lax-Friedrichs flux. According to (16) its ith component is defined by

$$\mathcal{H}_i(\mathbf{u}^+, \mathbf{u}^-, \mathbf{n}) = \frac{1}{2}\left(f^c_{ki}(\mathbf{u}^+)n_k + f^c_{ki}(\mathbf{u}^-)n_k + \alpha\left(\mathbf{u}^+, \mathbf{u}^-\right)\left(u_i^+ - u_i^-\right)\right).$$

Its derivative with respect to its first argument is then given by

$$\begin{aligned}\left(\mathcal{H}'_{\mathbf{u}^+}(\mathbf{u}^+, \mathbf{u}^-, \mathbf{n})\right)_{ij} &= \partial_{u_j^+}\mathcal{H}_i(\mathbf{u}^+, \mathbf{u}^-, \mathbf{n}) \\ &= \frac{1}{2}\left(\partial_{u_j} f^c_{ki}(\mathbf{u}^+)n_k + \alpha\delta_{ij} + \left(\alpha'_{\mathbf{u}^+}\left(\mathbf{u}^+, \mathbf{u}^-\right)\right)_j\left(u_i^+ - u_i^-\right)\right),\end{aligned}$$

where $\alpha = \alpha\left(\mathbf{u}^+, \mathbf{u}^-\right)$ and $\alpha'_{\mathbf{u}^+}\left(\mathbf{u}^+, \mathbf{u}^-\right)$ might be approximated by

$$\alpha'_{\mathbf{u}^+}\left(\mathbf{u}^+, \mathbf{u}^-\right) = \begin{cases} \operatorname{sign}\left(v_n(\mathbf{u}^+)\right)v'_n(\mathbf{u}^+) + c'(\mathbf{u}^+) & \text{for } \tilde{\alpha}(\mathbf{u}+) \geq \tilde{\alpha}(\mathbf{u}-), \\ 0 & \text{else,} \end{cases} \tag{46}$$

with $v_n(\mathbf{u}) = \mathbf{v}\cdot\mathbf{n}$ and $\tilde{a}(\mathbf{u}) = |v_n(\mathbf{u})| + c(\mathbf{u})$, see also (17) and (11). Similarly, we compute the derivative $\mathcal{H}'_{\mathbf{u}^-}(\mathbf{u}^+, \mathbf{u}^-, \mathbf{n})$ with respect to the second argument.

Finally, we consider the Vijayasundaram flux. According to (18) the ith component of this flux is given by

$$\mathcal{H}_i(\mathbf{u}^+, \mathbf{u}^-, \mathbf{n}) = B^+_{ij}(\hat{\mathbf{u}}, \mathbf{n})u_j^+ + B^-_{ij}(\hat{\mathbf{u}}, \mathbf{n})u_j^-,$$

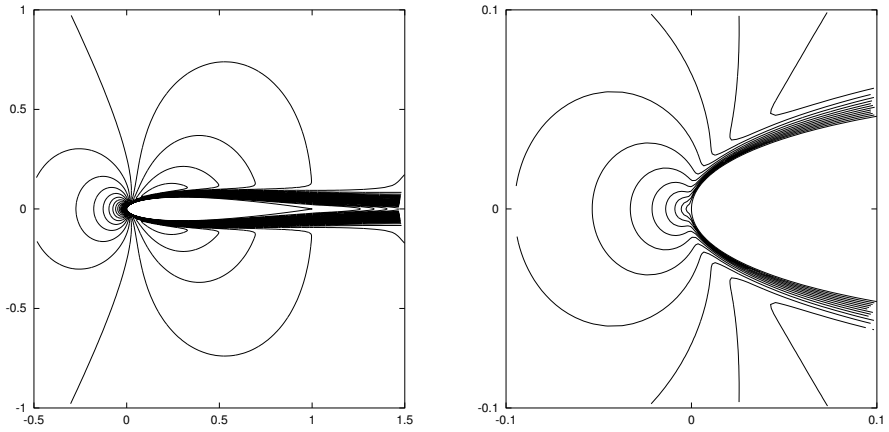
where the derivative with respect to the first argument is given by

$$\left(\mathcal{H}'_{\mathbf{u}^+}(\mathbf{u}^+, \mathbf{u}^-, \mathbf{n})\right)_{ij} = B^+_{ij}(\hat{\mathbf{u}}, \mathbf{n}) + \partial_{u_j^+}B^+_{ik}(\hat{\mathbf{u}}, \mathbf{n})u_k^+ + \partial_{u_j^+}B^-_{ik}(\hat{\mathbf{u}}, \mathbf{n})u_k^-.$$

Due to the involved dependence of $B^\pm(\hat{\mathbf{u}}, \mathbf{n})$ on $\mathbf{u}^+$ and $\mathbf{u}^-$, see the definition in (6), its derivative $\partial_{u_j^+}B^\pm_{ik}$ may be very complicated if computed explicitly. Instead, we approximate it by difference quotients, i.e.

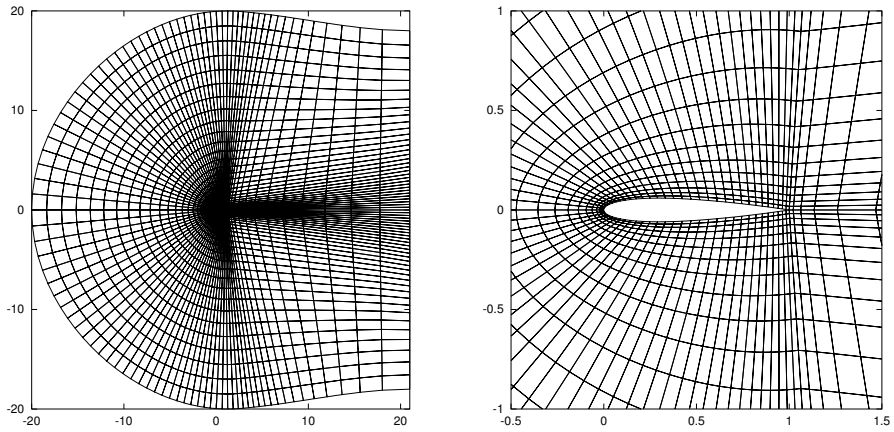
$$\begin{aligned}\partial_{u_k^+}B^\pm_{ij}(\bar{\mathbf{u}}, \mathbf{n}) &= \partial_{u_k^+}B^\pm_{ij}\left(\tfrac{1}{2}\left(\mathbf{u}^+ + \mathbf{u}^-\right), \mathbf{n}\right) \\ &= \tfrac{1}{2\varepsilon}\left(B^\pm_{ij}(\tfrac{1}{2}\left(\mathbf{u}^+ + \mathbf{u}^- + \varepsilon\mathbf{e}_k\right), \mathbf{n}\right) \\ &\qquad -B^\pm_{ij}(\tfrac{1}{2}\left(\mathbf{u}^+ + \mathbf{u}^- - \varepsilon\mathbf{e}_k\right), \mathbf{n})\Big) + \mathcal{O}\left(\varepsilon^2\right),\end{aligned} \tag{47}$$

where $\mathbf{e}_k$, $i = 1, 2$, are the unit vectors and $0 < \epsilon << 1$. Again, $\mathcal{H}'_{\mathbf{u}^-}(\mathbf{u}^+, \mathbf{u}^-, \mathbf{n})$ is computed accordingly.



(a) (b)

Figure 6: Mach isolines of the flow around the NACA0012 airfoil: (a) $\mathrm{Ma} = \frac{i}{50}, i = 1, \ldots$; (b) $\mathrm{Ma} = \frac{i}{20}, i = 1, \ldots$.



(a) (b)

Figure 7: Mesh: (a) Full view; (b) Zoom of coarse grid with 3072 elements.

5.3 Numerical performance of the Newton iteration

In this section we illustrate the performance of the Newton algorithm. To this end, we consider a subsonic viscous flow around a NACA0012 airfoil; here, the upper and lower surfaces of the airfoil geometry are specified by the function $g^{\pm}$, respectively, where

$$g^{\pm}(s) = \pm 5 \times 0.12 \times (0.2969 s^{1/2} - 0.126 s - 0.3516 s^2 + 0.2843 s^3 - 0.1015 s^4).$$

As the chord length l of the airfoil is $l \approx 1.00893$ we use a rescaling of g in order to yield an airfoil of unit (chord) length. The computational domain Ω is subdivided into quadrilateral elements; cf. the C-type grid depicted in Figures 7(a) and (b). Curved boundaries are approximated by piecewise quadratic polynomials. At the farfield (inflow) boundary we specify a Mach 0.5 flow at a zero angle of attack, with Reynolds number $\mathrm{Re} = 5000$; on the walls of the airfoil geometry, we impose a zero heat flux (adiabatic) no-slip boundary condition. This is a standard laminar test case which has been investigated by many other authors, cf. [6], for example. The solution to this problem consists of a strictly subsonic flow which is symmetric about the x-axis, see Figures 6(a) and (b).

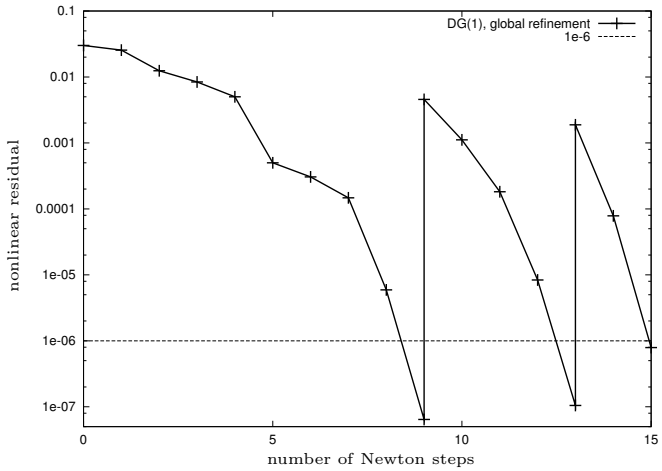


Figure 8: Convergence of the non-linear residual under global refinement with $p = 1$.

mesh	DG(1)	DG(2)	DG(3)
1	9	4*	4*
2	4	2	2
3	2	2	

Table 5: Number of Newton steps on each global refinement level for DG(p), $1 \leq p \leq 3$. (* pre-iteration on DG(1))

In the following we illustrate the performance of the Newton algorithm. on successively (globally) refined meshes for $p = 1, 2, 3$. The initial coarsest mesh consists of 3072 elements, cf. Figure 7; the two subsequently refined meshes have 12288 and 49152 elements, respectively. The starting guess on the coarsest mesh are taken to be the free-stream conditions. The linear problems arising in each Newton step are solved using GMRES and an ILU preconditioner included in the PETSc library, [3]. The ℓ_2-norm of the *linear* residual in each Newton step is reduced by a factor of 10^{-4}.

In Figure 8 we show the convergence history of the ℓ_2–norm of the *non-linear* residuals for $p = 1$. After seven damped Newton steps on the coarsest mesh the computed (approximate) solution is sufficiently close to the actual numerical solution so that the Newton algorithm proceeds without any damping, i.e. the damping parameter $\omega^n = 1$ and we observe quadratic-like Newton convergence. Once the ℓ_2–norm of the non-linear residual is below the prescribed tolerance (10^{-6}), the mesh is globally refined, and the solution is interpolated to the refined mesh. On the second and third meshes only four and two, respectively, undamped Newton steps are required until the convergence criterion is reached.

In Table 5 we summarize the number of Newton steps required to satisfy the nonlinear stopping criterion for both bilinear elements, as well as for higher–order elements with $p = 2, 3$. For higher–order elements, we use the numerical solution computed with bilinear elements on the coarsest mesh as the starting guess for the Newton algorithm. This then results in only 4 iterations being needed for both $p = 2, 3$ on the coarsest mesh; on the finer two meshes only two steps are required for both biquadratic and bicubic elements.

6 Higher order convergence of the DG method for the 2d compressible Navier-Stokes equations

In this section we demonstrate the computational accuracy of the higher order DG method derived in Section 4.2 for the 2d compressible Navier-Stokes equations. Concerning the availability of appropriate test cases, the situation for the 2d compressible Navier-Stokes is far more difficult than for the 2d compressible Euler equations. Whereas only few non-trivial problems governed by the compressible Euler equations with smooth solution are known, like e.g. the Ringleb flow problem shown in Section 3, there are virtually no *exact* (smooth) solutions known for the 2d compressible Navier-Stokes equations.

Given this, we will not be able to demonstrate the convergence of the higher order DG method for viscous flow with respect to the L^2 norm like in Section 3 for inviscid flows. Instead, in the following subsection, we consider the convergence of force coefficients for the case of a subsonic viscous flow around an airfoil. In the subsection thereafter, we concentrate on the higher order approximation of viscous boundary layers.

6.1 Subsonic flow around an airfoil: Convergence of force coefficients

In this section, we demonstrate the accuracy of the higher order DG method described in Section 4.2 for the computation of force coefficients. In particular, we consider the estimation of the drag and lift coefficients, c_{d} and c_{l}, respectively, which, in the case of a viscous flow, are defined by

$$J_{c_{\mathrm{d}}}(\mathbf{u}) = J_{c_{\mathrm{dp}}}(\mathbf{u}) + J_{c_{\mathrm{df}}}(\mathbf{u}),$$
$$J_{c_{\mathrm{l}}}(\mathbf{u}) = J_{c_{\mathrm{lp}}}(\mathbf{u}) + J_{c_{\mathrm{lf}}}(\mathbf{u}),$$

respectively, where c_{dp} and c_{lp} are the pressure induced force coefficients given by

$$J_{c_{\mathrm{dp}}}(\mathbf{u}) = \frac{2}{\bar{l}\bar{\rho}|\bar{\mathbf{v}}|^2}\int_S p\,(\mathbf{n}\cdot\psi_d)\,\mathrm{d}s, \qquad J_{c_{\mathrm{lp}}}(\mathbf{u}) = \frac{2}{\bar{l}\bar{\rho}|\bar{\mathbf{v}}|^2}\int_S p\,(\mathbf{n}\cdot\psi_l)\,\mathrm{d}s,$$

respectively, and c_{df} and c_{lf} are the viscous force coefficients, defined by

$$J_{c_{\mathrm{df}}}(\mathbf{u}) = \frac{2}{\bar{l}\bar{\rho}|\bar{\mathbf{v}}|^2}\int_S (\underline{\tau}\,\mathbf{n})\cdot\psi_d\,\mathrm{d}s, \qquad J_{c_{\mathrm{lf}}}(\mathbf{u}) = \frac{2}{\bar{l}\bar{\rho}|\bar{\mathbf{v}}|^2}\int_S (\underline{\tau}\,\mathbf{n})\cdot\psi_l\,\mathrm{d}s,$$

respectively. Here, S denotes the surface of the airfoil, $\bar{l}$ its chord length (equal to one), $\bar{\mathbf{v}}$ and $\bar{\rho}$ are the reference (or free-stream) velocity and density, respectively, $(\underline{\tau}\,\mathbf{n})\cdot\psi = \tau_{ij}n_j\psi_i$, where τ is the viscous stress tensor defined in (30) and

$$\psi_d = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad \psi_l = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

In this section, we consider the same test case as in Section 5.3, i.e. a Mach 0.5 flow at a zero angle of attack, with Reynolds number Re = 5000, around the NACA0012 airfoil with adiabatic no-slip boundary condition.

Given that this flow is symmetric about the x-axis the lift coefficients c_{lp} and c_{lf} are both zero. On the basis of fine grid computations, reference values of the drag coefficients are given by $J_{c_{\mathrm{dp}}}(\mathbf{u}) \approx 0.0222875$ and $J_{c_{\mathrm{df}}}(\mathbf{u}) \approx 0.032535$.

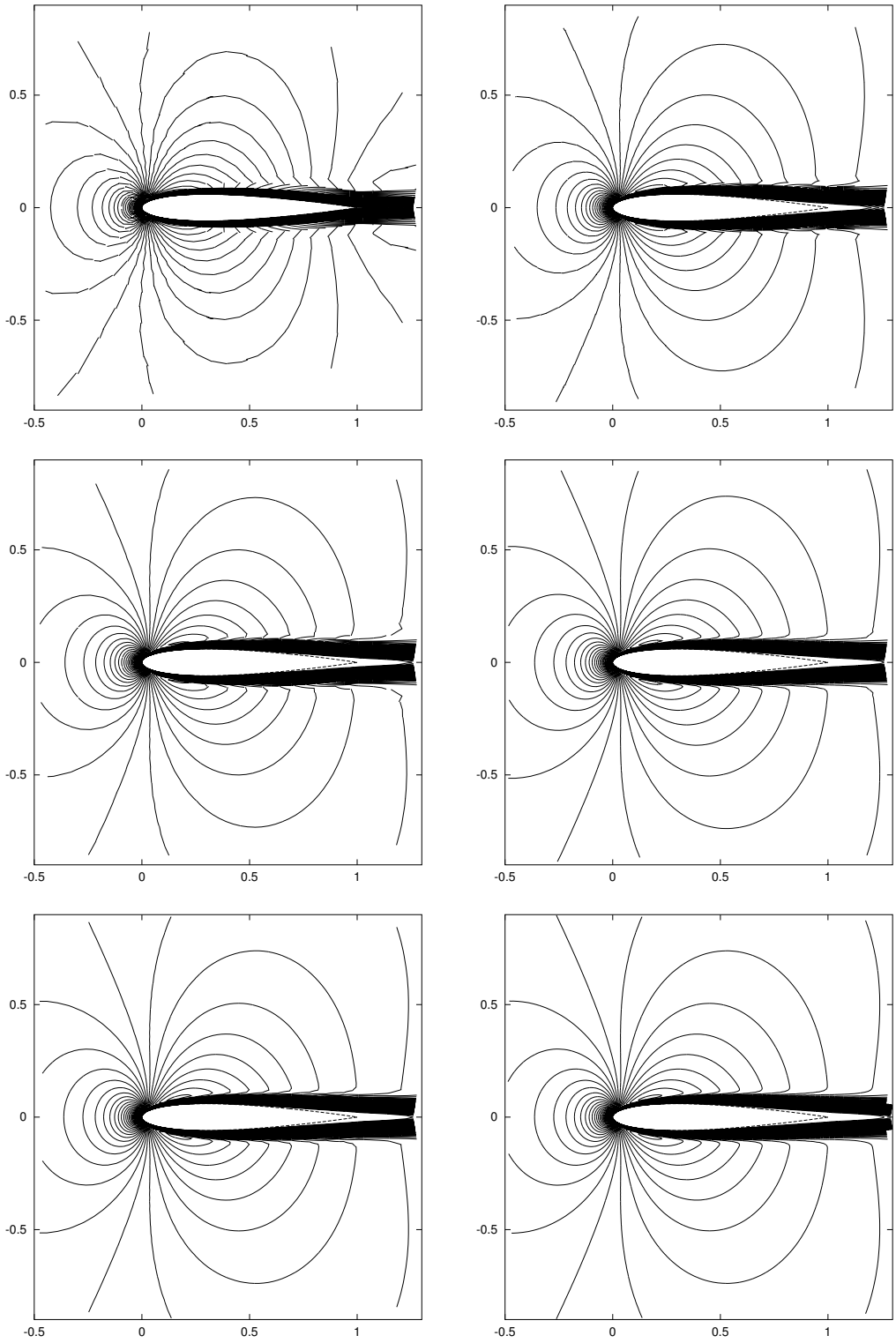


Figure 9: Mach isolines of the DG(p) solutions for (top) $p = 1$, (middle) $p = 2$, (bottom) $p = 3$ on (left) the coarsest mesh with 3072 elements and (right) on the mesh once refined with 12288 elements.

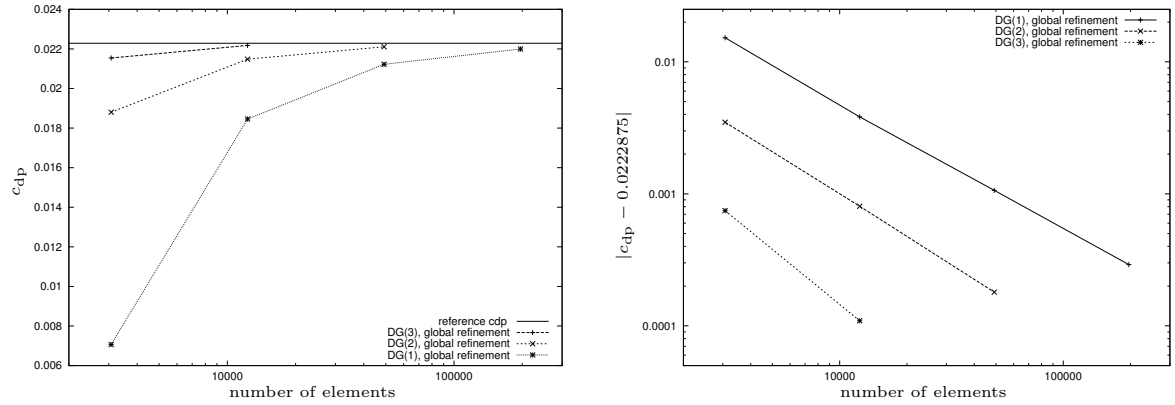


Figure 10: Convergence of $c_{\rm dp}$ under global refinement for DG(p), $p = 1, 2, 3$: (a) $c_{\rm dp}$ versus number of elements; (b) Error in $c_{\rm dp}$ (reference $c_{\rm dp} - c_{\rm dp}$) versus number of elements.

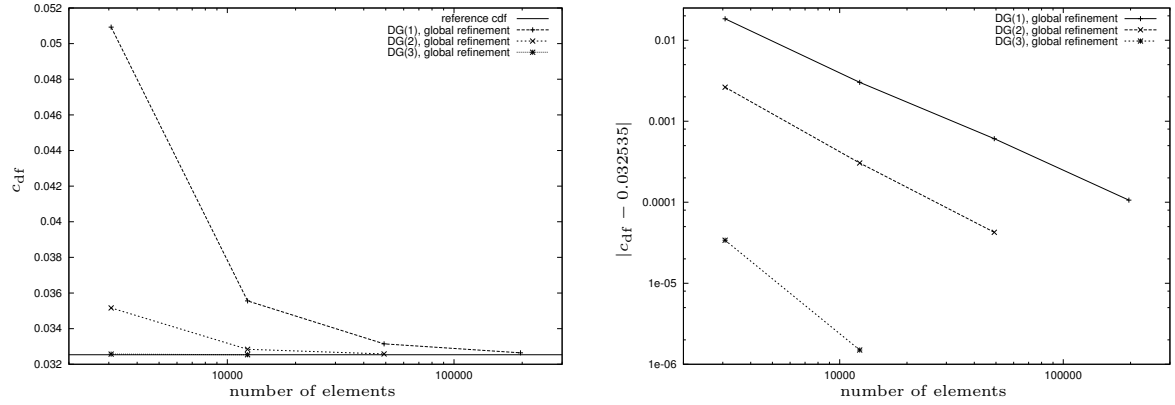


Figure 11: Convergence of $c_{\rm df}$ under global refinement for DG(p), $p = 1, 2, 3$: (a) $c_{\rm df}$ versus number of elements; (b) Error in $c_{\rm df}$ ($c_{\rm df}-$ reference $c_{\rm df}$) versus number of elements.

In Figure 9 we plot the Mach isolines of the computed numerical approximation using the interior penalty discontinuous Galerkin discretization on both the coarsest mesh and the mesh once refined for $p = 1, 2, 3$. Here, we clearly see that as the mesh is refined and the polynomial degree is increased, the quality of the numerical approximation significantly improves. Indeed, from Figures 10 and 11, we observe that the error in the approximation to both $c_{\rm dp}$ and $c_{\rm df}$, respectively, decreases when either the mesh is globally refined, or the polynomial degree is uniformly increased.

Finally, in Figure 12 we compare the error in the approximations to $c_{\rm dp}$ using global mesh refinement with an adaptive mesh refinement strategy employing residual-based error indicators together with the fixed fraction strategy (with refinement and derefinement fractions set to 20% and 10%, respectively), cf. [24], for details. We see from Figure 12 that the gain in accuracy in the approximation of $c_{\rm dp}$ using local refinement versus global refinement significantly increases as the polynomial degree is increased, which is expected for this smooth problem. Indeed, by employing local variation of the polynomial degree, depending on the local smoothness of the solution, in addition to

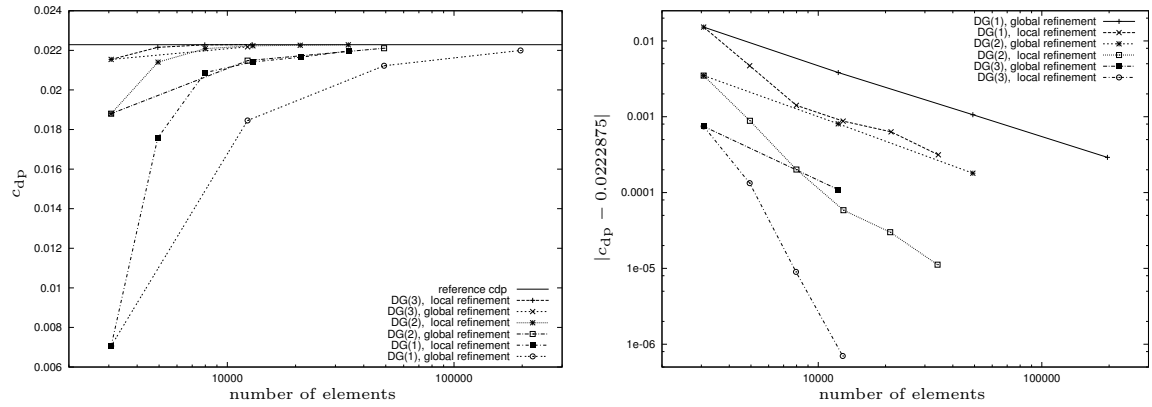


Figure 12: Convergence of c_{dp} under global mesh refinement and local mesh refinement using residual-based indicators.

local mesh refinement, i.e., *hp*-refinement, should be extremely efficient for this problem; this will be investigated as part of future research.

6.2 Flow over a flat plate

As we saw in the previous subsection, an increase in the polynomial degree leads to a dramatic improvement in the accuracy of the computed force coefficients c_{dp} and (even more pronounced for) c_{df}, cf. Figures 10 and 11, respectively. To a large extent this improvement in accuracy for these force coefficients can be attributed to the discretization's ability to accurately resolve the viscous boundary layer in the vicinity of the airfoil profile.

Thereby, in order to gain additional insight into how well the DG method can approximate boundary layers as the polynomial degree is increased, in this example we consider the flow over a flat plate. To this end, we consider a Mach 0.01 flow with Reynolds number 10000 horizontally passing over a flat plate of length $l = 2$. The boundary layer solution to this problem can be approximated using Blasius' solution, see [33], for example. In Figure 13, we compare the numerical solution computed with the DG method for $1 \leq p \leq 3$, at $x = \frac{l}{2} = 1$ and a local Reynolds number $Re_x = 5000$, with the Blasius solution ($\eta = y\sqrt{u_\infty/(\nu x)} = \frac{y}{x}\sqrt{Re_x}$ versus u/u_∞, cf. [33]) on a sequence of rather coarse computational meshes. On the coarsest mesh, which has about one or two elements within the boundary layer, we see that the DG solution computed with $p = 1, 2$ are not very close to the Blasius solution; increasing the polynomial order to $p = 3$ clearly yields a dramatic improvement in the underlying computed numerical solution. On the next finer mesh, where three elements are placed within the boundary layer, the bilinear approximation is still not very accurate, though now both the computed solution with $p = 2, 3$ are in excellent agreement with the Blasius solution. On the subsequent two meshes we clearly observe that the DG approximation with bilinear elements ($p = 1$) finally starts to coincide with the Blasius solution, at least on a macroscopic level. A more detailed view of the numerical solution on these latter two finer meshes is shown in the zoom depicted in Figure 14. Here, we see that there is still a significant difference between the Blasius solution and the computed discontinuous Galerkin solution with $p = 1$. Indeed, these figures clearly highlight the substantial gains in accuracy attained when higher–order polynomial degrees are employed with the DG method. This is fur-

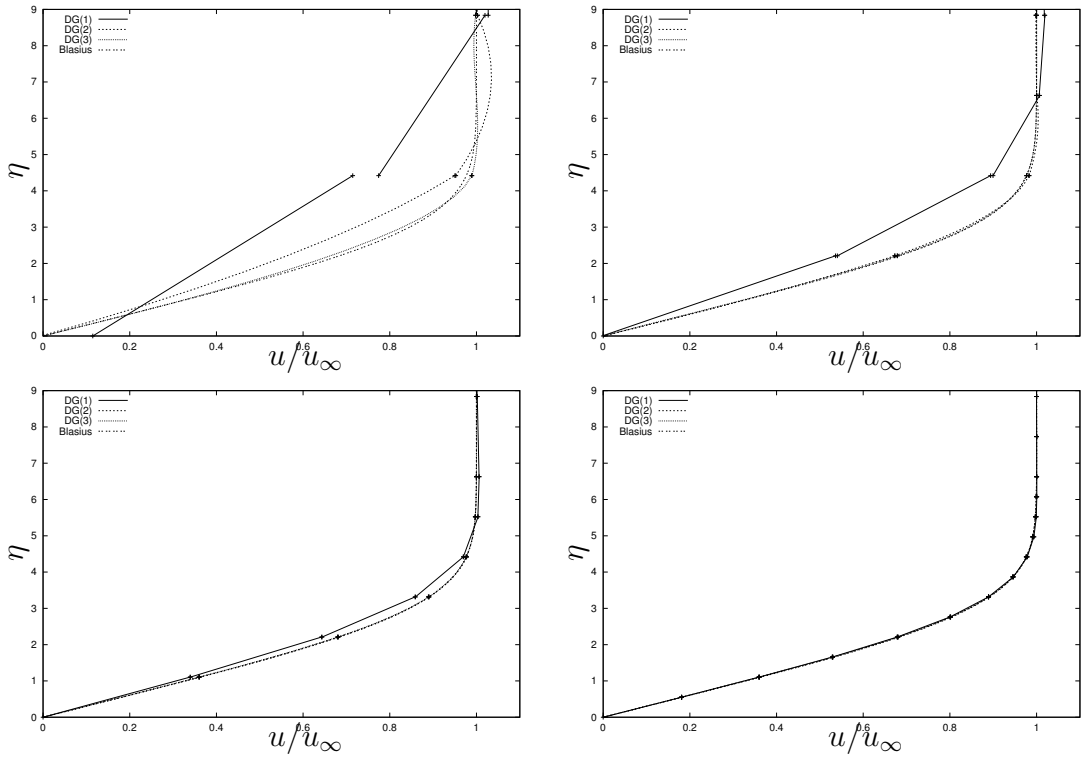


Figure 13: DG(p), $1 \leq p \leq 3$, solutions in comparison with the Blasius solution ($\eta = y\sqrt{u_\infty/(\nu x)} = \frac{y}{x}\sqrt{Re_x}$ versus u/u_∞) on a sequence of meshes with an increasing number of elements.

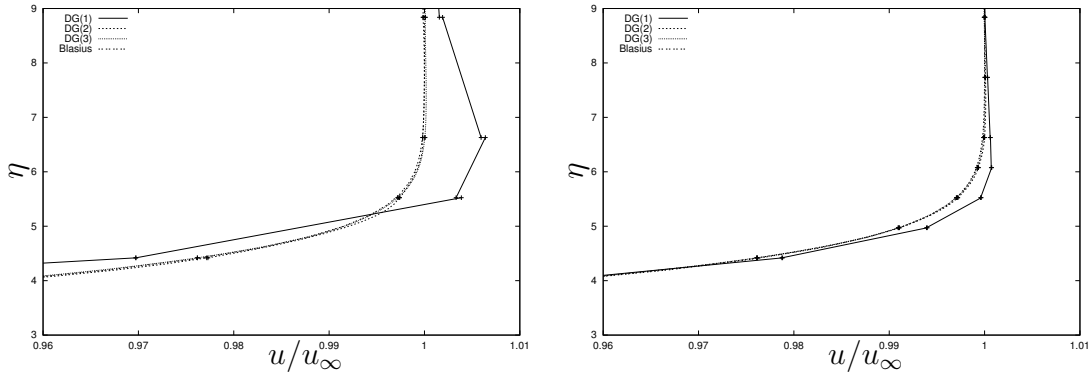


Figure 14: Zoom of the DG(p), $1 \leq p \leq 3$, solutions with the Blasius solution on the two finest grids.

	DG(1)	DG(2)	DG(3)
elements	36	5	3
DoFs	72	15	12

Table 6: Number of elements and degrees of freedom in the boundary layer required by DG(p), $1 \leq p \leq 3$, discretizations for approximating the viscous force up to 5%.

ther highlighted in Table 6, where we summarize the number of elements and the number of degrees of freedom, orthogonal to the wall, which are required by the DG method for each polynomial degree in order to resolve the boundary layer to a sufficient accuracy that the error in computed viscous stress forces exerted on the wall are within 5% of that computed with the Blasius solution.

7 Shock-capturing

7.1 Introduction

Discontinuous Galerkin schemes exhibit an inherent stability at discontinuities as can be seen, for example, when solving a linear advection equation with discontinuous boundary values imposed, or when solving compressible Navier-Stokes equations for problems including weak shocks on sufficiently coarse meshes. Indeed, these problems can be discretized and solved without any stabilization applied, although the resulting discrete solutions might suffer from spurious oscillations near the discontinuities. When these oscillations are to be suppressed, when nonlinear discontinuities as e.g. shocks are strong enough or when computing on sufficiently fine meshes, the discontinuous Galerkin discretizations must be stabilized.

Several stabilization techniques have been proposed: two of the most frequently used are i) (generalized) limiters as e.g. proposed by Cockburn and Shu, e.g. [15], in the framework of Runge-Kutta discontinuous Galerkin schemes, and ii) the addition of artificial viscosity terms, also referred to as shock-capturing (or discontinuity-capturing) terms.

As discussed in [36], local projection or slope limiters, such as the one proposed by Cockburn and Shu [15] have some disadvantages: one is attributed to serious problems which might occur when iterating a stationary solution to steady state. As the limited solution does not satisfy the steady-state discontinuous Galerkin equations, it is not possible to reduce the residual to machine accuracy, see [36]. Instead, the scheme tries to converge to the unlimited solution, which suffers from numerical oscillations and the limiter must remain active to prevent this.

An alternative stabilization approach is the use of artificial viscosity terms which were originally introduced by Hughes and Johnson in the context of SUPG and SD finite element methods and later also in DG methods for scalar hyperbolic conservation equations, see [26]. This approach adds an artificial viscosity term of the form

$$\sum_{\kappa} \int_{\kappa} \varepsilon \nabla \mathbf{u}_h : \nabla \mathbf{v}_h d\mathbf{x}, \tag{48}$$

to the discretization scheme, where κ are the elements which cover the domain Ω and $\mathbf{u}_h$ and $\mathbf{v}_h$ are discrete ansatz- and test functions taken from the finite element space $\mathbf{V}_h^p$. Most of these approaches differ in the specific choice of the coefficient ε, only. Several examples which have been employed for stabilizing discontinuous Galerkin discretizations of the stationary compressible Euler equations,

$$\nabla \cdot \mathcal{F}^c(\mathbf{u}) = 0, \tag{49}$$

near shocks will be given in the following.

Already in [5], Bassi and Rebay employed an artificial viscosity term of the form (48), where the scalar coefficient ε depends on the residual of the finite element solution $\mathbf{u}_h$ and on the diameter h_κ of the element κ, namely,

$$\varepsilon|_\kappa = C_\varepsilon h_\kappa^2 \left\{ \sum_i \left((|u_h^i| + c)^{-1} \left[\nabla \cdot \mathcal{F}^c(\mathbf{u}_h) \right]_i \right)^2 \right\}^{-\frac{1}{2}}, \tag{50}$$

where C_ε and c are positive parameters and i runs over all components of $\mathbf{u}$. Later, Bassi and Rebay, [8], used a viscosity term which depends on the face residuals instead of on the element residuals with the scalar coefficient ε given by

$$\varepsilon|_\kappa = C_\varepsilon \left\{ \sum_i \left[\frac{\int_{\partial\kappa} \mathcal{H}_i - \mathcal{F}_i^c(\mathbf{u}_h) \cdot \mathbf{n}_\kappa \, \mathrm{d}s}{\left(\int_\kappa u_h^i \, \mathrm{d}\mathbf{x} \right) / |\kappa|} \right]^2 \right\}^{-\frac{1}{2}}, \tag{51}$$

where $\mathbf{n}_\kappa$ denotes the unit outward normal vector to the boundary $\partial\kappa$ and $\mathcal{H}$ is a numerical flux function approximating the flux, $\mathcal{F}^c(\mathbf{u}_h) \cdot \mathbf{n}_\kappa$, at element interfaces taking into account possible discontinuities of $\mathbf{u}_h$ across the interfaces, see Section 2.4 for more detail.

Also, Baumann and Oden, [10], employed an artificial viscosity term like in (48), with a scalar coefficient given by

$$\varepsilon|_\kappa = h_\kappa \lambda_\kappa(\mathbf{u}_h), \tag{52}$$

on elements κ close to sharp gradients, only, and zero elsewhere. Here, $\lambda_\kappa(\mathbf{u}_h)$ is the maximum characteristic speed, taken as $c + |\mathbf{v}|$, where c is the speed of sound and $\mathbf{v}$ the velocity vector.

Finally, also in the framework of space-time discontinuous Galerkin schemes, [36], artificial viscosity stabilization terms have been used, one of which is closely related to (50).

In the current publication we employ an artificial viscosity term, (48), with a coefficient similar to

$$\varepsilon|_\kappa = C_\varepsilon \, h_\kappa^{2-\boldsymbol{\beta}} \, |\mathbf{R}(\mathbf{u}_h)|, \tag{53}$$

with C and $\boldsymbol{\beta}$ positive constants and $\mathbf{R}(\mathbf{u}) = -\nabla \cdot \mathcal{F}^c(\mathbf{u})$. This is a slight simplification of the artificial viscosity term proposed in [26] and has already been successfully applied in [22] to the adaptive discontinuous Galerkin discretization for the solution of the compressible Euler equations. In fact, in the current publication not the isotropic version (53) is proposed, but a generalization of it for anisotropic meshes.

Like all artificial viscosity terms of the form (48) this stabilization leaves the conservativity property of the discontinuous Galerkin method unchanged. The coefficient ε in (53) is relatively simple in comparison to the coefficients in (50) and (51), which is useful when the scheme is to be treated implicitely. Furthermore, the coefficient $\varepsilon = \varepsilon(\mathbf{u})$ in (53) is continuous with respect to its argument which helps in the nonlinear solution iteration procedure. In particular, it does not include a switch which locally enables or disables the shock-capturing due to e.g. sharp gradients as does the method proposed in [9]. Finally, the artificial viscosity term (48) with the specific choice of the ε in (53) is consistent in the sense that it vanishes when evaluated for the exact and sufficiently regular solution $\mathbf{u}$ of equation (49). This results in an artificial viscosity acting only in non-smooth parts of the solution where the residuals are large, and almost vanishing in smooth parts of the solution where the residuals are significantly smaller. Finally, the

consistency of the artificial viscosity term ensures that the (local and global) Galerkin orthogonality of the discontinuous Galerkin scheme still holds after its addition. This is particularly important in the framework of *a posteriori* error estimation and adaptivity, see Section 8.

7.2 Shock-capturing for the compressible Navier-Stokes equations

In this section we define a shock-capturing term $\mathcal{N}_{sc}(\mathbf{u}_h, \mathbf{v}_h)$ which is added to the DG discretization (44) of the compressible Navier-Stokes equations.

As already indicated in Section 7.1, the discontinuous Galerkin discretization (44) for the compressible Navier-Stokes equations is supplemented with an artificial viscosity term given by

$$\mathcal{N}_{sc}(\mathbf{u}_h, \mathbf{v}_h) \equiv \sum_{\kappa} \int_{\kappa} \varepsilon(\mathbf{u}_h) \nabla \mathbf{u}_h : \nabla \mathbf{v}_h d\mathbf{x} \equiv \sum_{\kappa} \int_{\kappa} \varepsilon_{ki}(\mathbf{u}_h) \partial_{x_i} u_h^k \, \partial_{x_i} v_h^k \, d\mathbf{x}. \tag{54}$$

In particular, we choose the coefficient matrix ε_{ki} to be

$$\varepsilon_{ki}(\mathbf{u}_h) = C_\varepsilon \, h_i^{2-\boldsymbol{\beta}} \, \mathcal{R}_k(\mathbf{u}_h), \quad i = 1, 2, \;\; k = 1, \ldots, 4, \tag{55}$$

where C_ε and $\boldsymbol{\beta}$ are positive constants and h_i represents the dimension of element κ in the ith coordinate direction, $i = 1, 2$. Finally, $\mathcal{R}_k(\mathbf{u}_h)$, $k = 1, \ldots, 4$, is defined by

$$\mathcal{R}_k(\mathbf{u}) = \sum_{q=1}^{4} |R_q(\mathbf{u})|, \quad k = 1, \ldots, 4, \tag{56}$$

where $\mathbf{R}(\mathbf{u}_h) = (R_q(\mathbf{u}_h), q = 1, \ldots, 4)$ denotes the residual of equations (28), i.e.

$$\mathbf{R}(\mathbf{u}) = -\nabla \cdot \left(\mathcal{F}^c(\mathbf{u}) - \mathcal{F}^v(\mathbf{u}, \nabla \mathbf{u}) \right). \tag{57}$$

Remarks.

(i) Due to the specific choice of the artificial viscosity term $\mathcal{N}_{sc}(\mathbf{u}_h, \mathbf{v}_h)$ depending on the residual $\mathbf{R}(\mathbf{u})$, see (57), this term vanishes when evaluated for the exact and sufficiently regular solution $\mathbf{u}$ to the problem (28). This ensures that the discretization remains consistent in the sense that the Galerkin orthogonality,

$$\mathcal{N}(\mathbf{u}, \mathbf{v}_h) - \mathcal{N}(\mathbf{u}_h, \mathbf{v}_h) = 0 \quad \mathbf{v}_h \in \mathbf{V}_h^p \tag{58}$$

with $\mathbf{u}$ and $\mathbf{u}_h$ denoting the solutions of (28) and (44), respectively, which is valid for the discretization without shock-capturing, is still valid when the shock-capturing term $\mathcal{N}_{sc}(\cdot, \cdot)$ is included.

(ii) Furthermore, we note that in (56), $\mathcal{R}_k(\mathbf{u})$ is the same constant for all components k, $k = 1, \ldots, 4$. A simpler choice,

$$\mathcal{R}_k(\mathbf{u}) = R_k(\mathbf{u}), \quad k = 1, \ldots, 4,$$

is cheaper in terms of assembling time of the Jacobian, cf. Section 5, but turned out to be unstable for various numerical test cases.

(iii) The choice of the coefficient matrix in (55) represents an extension of the shock-capturing term, already employed in [22] for the compressible Euler equations, to the compressible Navier-Stokes equations. Furthermore, it represents a generalization to anisotropic meshes. In fact, its isotropic version, i.e. setting $h_i = h_\kappa$, $i = 1, 2$, where h_κ represents the diameter of the element κ, was found to not work on meshes with anisotropic elements.

7.3 The Jacobian of the shock-capturing term

As described in Section 5 the system of nonlinear equations is solved using a Newton–GMRES algorithm. Key ingredient of this iteration method is the linear system (45) which includes the derivative (or Jacobian) $\mathcal{N}'_{\mathbf{u}}[\mathbf{w}](\boldsymbol{\phi}, \mathbf{v})$ of the discretization scheme $\mathbf{u} \to \mathcal{N}(\mathbf{u}, \mathbf{v})$. A detailed description of how $\mathcal{N}'_{\mathbf{u}}[\mathbf{w}](\cdot, \mathbf{v})$ is approximated for the DG discretization (44) is given in Section 5. In addition to this, the Jacobian $\mathcal{N}'_{sc,\mathbf{u}}[\mathbf{w}](\boldsymbol{\phi}, \mathbf{v})$ of the shock-capturing term $\mathcal{N}_{sc}(\cdot, \mathbf{v})$ described in the previous section can be approximated as follows

$$\hat{\mathcal{N}}'_{sc,\mathbf{u}}[\mathbf{w}](\boldsymbol{\phi}, \mathbf{v}) = \sum_\kappa \int_\kappa \varepsilon(\mathbf{w}) \nabla\boldsymbol{\phi} \cdot \nabla\mathbf{v} d\mathbf{x} + \sum_\kappa \int_\kappa \hat{\varepsilon}'_{\mathbf{u}}[\mathbf{w}](\boldsymbol{\phi}) \nabla\mathbf{w} \cdot \nabla\mathbf{v} d\mathbf{x}, \tag{59}$$

where $\hat{\varepsilon}'_{\mathbf{u}}[\mathbf{w}](\boldsymbol{\phi})$ is given by

$$\hat{\varepsilon}'_{ik,\mathbf{u}}[\mathbf{w}](\boldsymbol{\phi}) = C_\varepsilon \, h_k^{2-\boldsymbol{\beta}} \, \mathcal{R}'_{i,\mathbf{u}}[\mathbf{w}](\boldsymbol{\phi}), \quad i = 1, \ldots, 4, \ k = 1, 2,$$

and

$$\mathcal{R}'_{i,\mathbf{u}}[\mathbf{w}](\boldsymbol{\phi}) = \sum_{q=1}^{4} \operatorname{sgn}(R_q(\mathbf{w})) R'_{q,\mathbf{u}}[\mathbf{w}](\boldsymbol{\phi}), \quad i = 1, \ldots, 4.$$

Recalling the definition of $R_q(\mathbf{u})$ in (57),

$$\begin{aligned} R_q(\mathbf{u}) &= -\partial_{x_p} f^c_{pq}(\mathbf{u}) + \partial_{x_p} f^v_{pq}(\mathbf{u}, \nabla\mathbf{u}) \\ &= -\partial_{u^r} f^c_{pq}(\mathbf{u}) \partial_{x_p} u^r + \partial_{u^r} \left(G(\mathbf{u})_{pl}\right)_{qs} \partial_{x_p} u^r \partial_{x_l} u^s + \left(G(\mathbf{u})_{pl}\right)_{qs} \partial_{x_p} \partial_{x_l} u^s, \end{aligned}$$

we obtain the following expression for $R'_{q,\mathbf{u}}[\mathbf{w}](\boldsymbol{\phi})$,

$$\begin{aligned} R'_{q,\mathbf{u}}[\mathbf{w}](\boldsymbol{\varphi}) = &- \partial_{u^j} f^c_{pq}(\mathbf{w}) \partial_{x_p} \varphi^j - \partial_{u^j} \partial_{u^r} f^c_{pq}(\mathbf{w}) \varphi^j \partial_{x_p} w^r \\ &+ \partial_{u^j} \partial_{u^r} \left(G(\mathbf{w})_{pl}\right)_{qs} \varphi^j \partial_{x_p} w^r \partial_{x_l} w^s \\ &+ \partial_{u^j} \left(G(\mathbf{w})_{pl}\right)_{qs} \partial_{x_p} \varphi^j \partial_{x_l} w^s \\ &+ \partial_{u^r} \left(G(\mathbf{w})_{pl}\right)_{qj} \partial_{x_p} w^r \partial_{x_l} \varphi^j \\ &+ \partial_{u^j} \left(G(\mathbf{w})_{pl}\right)_{qs} \varphi^j \partial_{x_p} \partial_{x_l} w^s \\ &+ \left(G(\mathbf{w})_{pl}\right)_{qj} \partial_{x_p} \partial_{x_l} \varphi^j. \end{aligned} \tag{60}$$

Remark. Evaluating the term including the second derivatives of G in the second line of (60) is extremely time consuming. In [20] it has been demonstrated that neglecting this term in the assembly of the Jacobian matrix does not deteriorate the convergence of the Newton algorithm. In fact, the same number of Newton steps are required than when this term is included. But, the time savings in the matrix assembly when neglecting this term, finally lead to a significant decrease of the overall computing time.

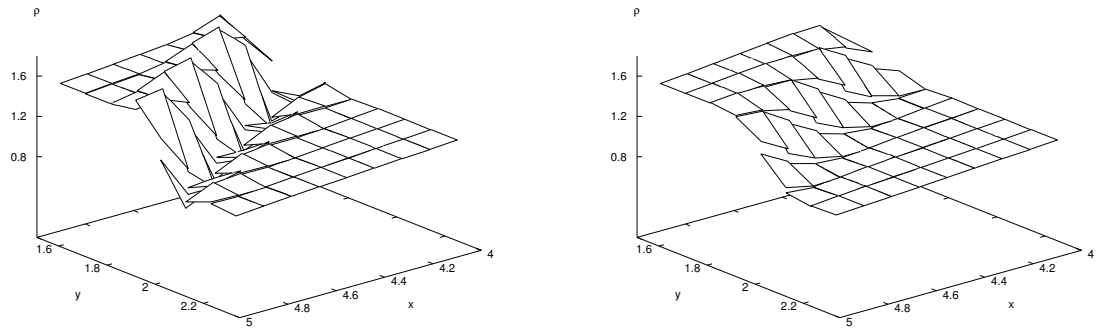


Figure 15: Supersonic flow past a wedge: (left) without and (right) with shock-capturing.

7.4 Numerical results

7.4.1 Supersonic flow past a wedge

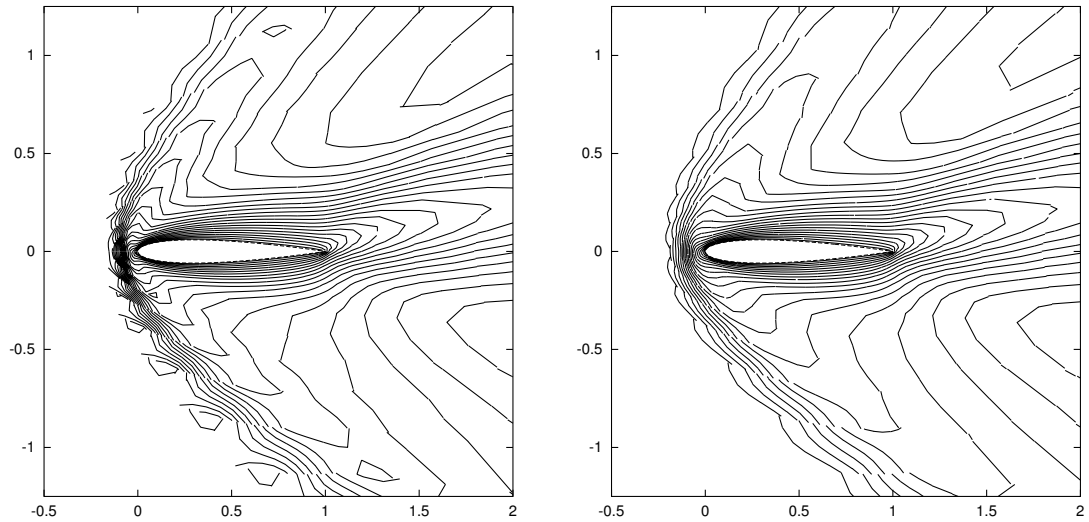
In the following, we demonstrate the effect of the shock–capturing method. To this end, we consider a supersonic flow past a wedge (compression corner problem), see Section 8.6.2 for more details.

The solution to this problem develops an oblique shock originating at the corner. In Figure 15, the shock at the outflow boundary is shown in detail for the numerical solution with and without shock–capturing. We note that in both cases the shock is resolved by 3-4 elements. Hence, the shock is not smeared by the shock-capturing but it is resolved by at most as many elements as for the case when no shock–capturing is employed.

7.4.2 Supersonic flow around an airfoil

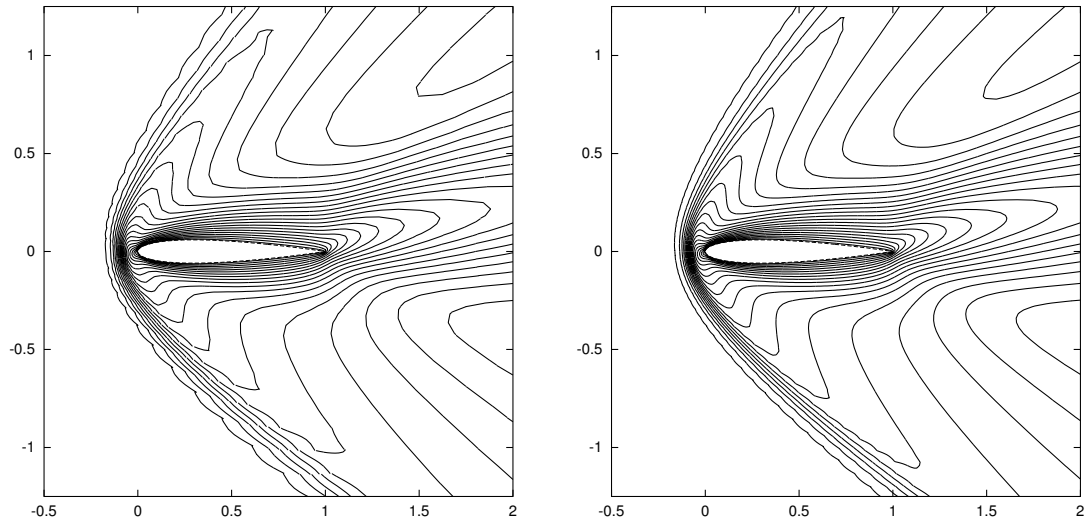
In this second example we consider a laminar flow at $M = 2$, $Re = 106$ and $\alpha = 10^\circ$ with constant temperature on the profile, a test case previously being considered in [6, 13], for example.

We note, that the discontinuous Galerkin discretization applied to this problem can be solved without any shock capturing employed as long as the numerical dissipation is sufficiently large, like on very coarse meshes, for example. In fact, in [6] this problem has been solved without any stabilization on a rather coarse mesh (O-grid with 1024 elements). Also on the mesh, see Figure 7, with 3072 elements, the unstabilized DG discretization can be solved, see Figure 16 for a comparison of the numerical solutions both with and without shock-capturing. But, when this mesh is once globally refined, the unstabilized version cannot be solved any more. In contrast to this, the stabilized version can be solved, see Figure 17(a) and (b) for the discrete solutions on the mesh depicted in Figure 7 being once and twice globally refined, respectively.



(a) (b)

Figure 16: $M = 2, \mathrm{Re} = 106, \alpha = 10^\circ$ flow around the NACA0012 airfoil: Mach isolines of the discrete solution (a) without and (b) with shock capturing on the mesh of 3072 elements shown in Figure 7.



(a) (b)

Figure 17: $M = 2, \mathrm{Re} = 106, \alpha = 10^\circ$ flow around the NACA0012 airfoil: Mach isolines of the discrete solution with shock-capturing on mesh Figure 7, which is (a) once and (b) twice globally refined.

8 Error estimation and adaptivity

8.1 Introduction

In aerodynamical computations like compressible flows around airfoils, much emphasis is placed on the accurate approximation of specific target quantities $J(\cdot)$, in particular, the force coefficients, see Section 6.1, like the pressure induced as well as the viscous stress induced drag, lift and moment coefficients, respectively. While local mesh refinement is required for obtaining reasonably accurate results in applications, the goal of the adaptive refinement is either to compute these coefficients as accurate as possible within given computing resources or to compute these coefficients up to a given tolerance with the minimum computing resources required. In both cases a goal-oriented refinement is needed, i.e. an adaptive refinement strategy specifically targeted to the efficient computation of the quantities of interest. Futhermore, in the latter case, an estimate is required of how accurate the force coefficients are approximated, i.e. an *a posteriori* error estimate is required of the error of the numerical solution measured in terms of the quantity of interest.

This error can be represented by the element and face residuals of the primal (flow) solution multiplied by the solution of a dual (adjoint) problem with data coupling to the specific target quantity. By approximating the solution to the dual problem numerically, the resulting approximate error representation gives an estimate of the true error. Furthermore, the approximate error representation can be decomposed as a sum over all elements of *adjoint-based* (also called *dual-weighted residual*) indicators which can be used for goal-oriented (adjoint-based) refinement specifically tailored to the efficient computation of the quantities of interest.

The approach of *a posteriori* error estimation and adaptivity in finite element methods has been developed in [11] and applied to various kinds of problems, see the survey article [12].

In [18], this approach has been developed for the discontinuous Galerkin discretization of scalar hyperbolic problems. Then, in the series of publications, [19, 21, 22], it has been extended to the two–dimensional compressible Euler equations, where a variety of problems have been considered, including the Ringleb flow problem, supersonic flow past a wedge, inviscid flows through a nozzle, and inviscid sub-, trans- and supersonic flows around different airfoil geometries; finally, in [23] and [24], this approach has been extended to the two–dimensional compressible Navier-Stokes equations and applied to subsonic viscous compressible flows around simple airfoil geometries. [20] gives the extension of this approach to viscous compressible flows including shocks, like supersonic flows, for example.

In the following subsection we will define the *dual* problem, also called *adjoint* problem, which is required in the *a posteriori* error estimation and the related adaptive mesh refinement algorithm. For scalar hyperbolic problems and various target quantities we explicitely give the adjoint equations in Section 8.3. After deriving the *a posteriori* error estimation in Section 8.4 and describing the adaptive algorithm in Section 8.5, we give several numerical examples in Section 8.6. In particular, we show the adjoint solutions for various problems, we demonstrate the accuracy of the error estimation as well as the performance of the adaptive refinement algorithm.

8.2 The dual (adjoint) problem

Assuming that the functional of interest $J(\cdot)$ is differentiable, we write $\bar{J}(\cdot;\cdot)$ to denote the mean value linearization of $J(\cdot)$ defined by

$$\bar{J}(\mathbf{u},\mathbf{u}_h;\mathbf{u}-\mathbf{u}_h) = J(\mathbf{u}) - J(\mathbf{u}_h) = \int_0^1 J'[\theta\mathbf{u}+(1-\theta)\mathbf{u}_h](\mathbf{u}-\mathbf{u}_h)\,\mathrm{d}\theta, \tag{61}$$

where $J'[\mathbf{w}](\cdot)$ denotes the Fréchet derivative of $J(\cdot)$ evaluated at some $\mathbf{w}$ in $\mathbf{V}$. Analogously, we write $\mathcal{M}(\mathbf{u},\mathbf{u}_h;\cdot,\cdot)$ to denote the mean–value linearization of $\mathcal{N}(\cdot,\cdot)$ given by

$$\begin{aligned}\mathcal{M}(\mathbf{u},\mathbf{u}_h;\mathbf{u}-\mathbf{u}_h,\mathbf{v}) &= \mathcal{N}(\mathbf{u},\mathbf{v}) - \mathcal{N}(\mathbf{u}_h,\mathbf{v}) \\ &= \int_0^1 \mathcal{N}'_{\mathbf{u}}[\theta\mathbf{u}+(1-\theta)\mathbf{u}_h](\mathbf{u}-\mathbf{u}_h,\mathbf{v})\,\mathrm{d}\theta \end{aligned}\tag{62}$$

for all $\mathbf{v}$ in $\mathbf{V}$. Here, $\mathcal{N}'_{\mathbf{u}}[\mathbf{w}](\cdot,\mathbf{v})$ denotes the Fréchet derivative of $\mathbf{u}\mapsto\mathcal{N}(\mathbf{u},\mathbf{v})$, as defined in Section 5. We now introduce the following *dual* or *adjoint* problem: find $\mathbf{z}\in\mathbf{V}$ such that

$$\mathcal{M}(\mathbf{u},\mathbf{u}_h;\mathbf{w},\mathbf{z}) = \bar{J}(\mathbf{u},\mathbf{u}_h;\mathbf{w}) \quad \forall\mathbf{w}\in\mathbf{V}. \tag{63}$$

We assume that (63) possesses a unique solution. Clearly, the validity of this assumption depends on both the definition of $\mathcal{M}(\mathbf{u},\mathbf{u}_h;\cdot,\cdot)$ and the choice of the target functional under consideration. In the following we must therefore *assume* that the dual problem (63) is well–posed, see the following subsection for cases where well-posedness is known.

As in most cases the exact solution $\mathbf{z}$ to the dual problem (63) is not known, it is approximated numerically. However, (63) includes the unkown *exact* solution $\mathbf{u}$ to the primal problem. Thus, in order to approximate the dual solution $\mathbf{z}$, we must replace $\mathbf{u}$ in (63) by a suitable approximations. The linearizations leading to $\mathcal{M}(\mathbf{u},\mathbf{u}_h;\cdot,\cdot)$ and $\bar{J}(\mathbf{u},\mathbf{u}_h;\cdot)$ are performed about $\mathbf{u}_h$, resulting in $\mathcal{N}'[\mathbf{u}_h](\cdot,\cdot)$ and $J'[\mathbf{u}_h](\cdot)$, respectively. The *linearized dual problem*: find $\hat{\mathbf{z}}\in\mathbf{V}$ such that

$$\mathcal{N}'[\mathbf{u}_h](\mathbf{w},\hat{\mathbf{z}}) = J'[\mathbf{u}_h](\mathbf{w}) \quad \forall\mathbf{w}\in\mathbf{V}, \tag{64}$$

is then discretized using the discontinuous Galerkin method, to yield following *approximate dual problem*: find $\hat{\mathbf{z}}_h\in\mathbf{V}_h^{\hat{p}}$ such that

$$\hat{\mathcal{N}}'[\mathbf{u}_h](\mathbf{w}_h,\hat{\mathbf{z}}_h) = J'[\mathbf{u}_h](\mathbf{w}_h) \quad \forall\mathbf{w}_h\in\mathbf{V}_h^{\hat{p}}. \tag{65}$$

In common with each Newton iteration step (45), the linearized and discretized dual problem (65) includes the *Jacobian* of the scheme. In fact, (65) represents a linear problem with a matrix which is the transpose of the matrix of a Newton iteration step, though assembled on $\mathbf{V}_h^{\hat{p}}$ which might be different from $\mathbf{V}_h^p$, see (70).

Remark. Equation (65) represents the so-called *discrete* adjoint problem. The *discrete adjoint approach* starts with the discretization of the nonlinear primal equations, takes its linearization and forms its transpose. In contrast to that, the *continuous adjoint approach* takes the adjoint equations of the nonlinear primal equations, which are then discretized.

The following subsection gives several adjoint problems for a scalar hyperbolic problem. For these linear examples the adjoint problems based on the discrete and continuous adjoint approach coincide.

8.3 Adjoint problems for scalar hyperbolic problems

We consider the case of the scalar linear hyperbolic equation (19)

$$\begin{aligned} \boldsymbol{\beta} \cdot \nabla u + bu &= f \quad \text{in } \Omega, \\ u &= g \quad \text{on } \Gamma_-. \end{aligned} \tag{66}$$

In the following we collect some linear target functionals $J(\cdot)$ for that the dual (adjoint) problem defined in (63) is known to be well–posed:

1. *Outflow normal flux:* Given a weight function $\psi \in L^2(\Gamma_+)$ we consider the weighted normal flux through the outflow boundary Γ_+ that is defined by

$$J(w) = \int_{\Gamma_+} \boldsymbol{\beta} \cdot \mathbf{n} w \psi \, ds.$$

   Then, z is the unique solution to the following boundary value problem: find $z \in V$ such that

$$\begin{aligned} -\nabla \cdot (\boldsymbol{\beta} z) + bz &= 0 \quad \text{in } \Omega, \\ z &= \psi \quad \text{on } \Gamma_+. \end{aligned}$$

2. *Mean value:* Here, let ψ be a weight function in $L^2(\Omega)$. Then a weighted mean value is given by

$$J(w) = \int_{\Omega} w \psi \, ds.$$

   In this case, z is the solution to following dual problem: find $z \in V$ such that

$$\begin{aligned} -\nabla \cdot (\boldsymbol{\beta} z) + bz &= \psi \quad \text{in } \Omega, \\ z &= 0 \quad \text{on } \Gamma_+. \end{aligned}$$

3. *Point value:* Finally we consider a point value

$$J(w) = w(x_0)$$

   at a given point $x_0 \in \Omega$. The existence and uniqueness of a dual solution corresponding to this non–regular target functional can be shown by first considering regularized target functionals $J_\epsilon(w) = \int_\Omega w \psi_{x_0,\epsilon} \, ds$ with $\psi_{x_0,\epsilon} = \epsilon^{-d} \phi((x - x_0)/\epsilon)$ where ϕ denotes a nonnegative function that is zero outside the unit ball. For $\epsilon \to 0$ this results in a unique solution z to the following dual problem

$$\begin{aligned} -\nabla \cdot (\boldsymbol{\beta} z) + bz &= \delta_{x_0} \quad \text{in } \Omega, \\ z &= 0 \quad \text{on } \Gamma_+, \end{aligned}$$

   where here, δ_{x_0} denotes a δ-distribution at point x_0 with the property

$$\int_{\Omega} \delta_{x_0} w \, ds = w(x_0).$$

8.4 Error estimation

Assuming that the dual problem (63) is well–posed, we have the following result.

Theorem 8.1 *Let* $\mathbf{u}$ *and* $\mathbf{u}_h$ *denote the solutions of* (28) *and* (44)*, respectively, and suppose that the dual problem* (63) *is well–posed. Then,*

$$J(\mathbf{u}) - J(\mathbf{u}_h) = -\mathcal{N}(\mathbf{u}_h, \mathbf{z} - \mathbf{z}_h), \tag{67}$$

for all $\mathbf{z}_h$ *in* $\mathbf{V}_h^p$.

Proof: Choosing $\mathbf{w} = \mathbf{u} - \mathbf{u}_h$ in (63), recalling the linearization performed in (61), and exploiting the Galerkin orthogonality property (58) for all $\mathbf{v}_h$ in $\mathbf{V}_h^p$, we get

$$\begin{aligned} J(\mathbf{u}) - J(\mathbf{u}_h) &= \bar{J}(\mathbf{u}, \mathbf{u}_h; \mathbf{u} - \mathbf{u}_h) = \mathcal{M}(\mathbf{u}, \mathbf{u}_h; \mathbf{u} - \mathbf{u}_h, \mathbf{z}) \\ &= \mathcal{M}(\mathbf{u}, \mathbf{u}_h; \mathbf{u} - \mathbf{u}_h, \mathbf{z} - \mathbf{z}_h) = -\mathcal{N}(\mathbf{u}_h, \mathbf{z} - \mathbf{z}_h) \end{aligned}$$

for all $\mathbf{z}_h$ in $\mathbf{V}_h^p$.

Based on the general error representation formula derived in Theorem 8.1, which can be written as follows,

$$J(\mathbf{u}) - J(\mathbf{u}_h) = -\mathcal{N}(\mathbf{u}_h, \mathbf{z} - \mathbf{z}_h) \equiv \sum_{\kappa \in \mathcal{T}_h} \eta_\kappa, \tag{68}$$

where η_κ includes the face and element residuals multiplied by the dual solution, see [24], *a posteriori* error estimates bounding the error in the computed functional $J(\cdot)$ may be deduced. Under the assumptions of Theorem 8.1, we have

$$|J(\mathbf{u}) - J(\mathbf{u}_h)| \leq \sum_{\kappa \in \mathcal{T}_h} |\eta_\kappa| \,. \tag{69}$$

This bound follows from (68) by application of the triangle inequality.

We end this section by noting that both the error representation formula (67) and the *a posteriori* error bound (69) depend on the unknown analytical solution $\mathbf{z}$ to the dual problem (63). Thus, in order to render these quantities computable, $\mathbf{z}$ is replaced by the solution $\hat{\mathbf{z}}_h$ to the approximate dual problem (65). Here, the approximate dual solution $\hat{\mathbf{z}}_h$ is computed on the same mesh $\mathcal{T}_h$ used for $\mathbf{u}_h$, but with a higher degree polynomial, i.e., $\hat{\mathbf{z}} \in \mathbf{V}_h^{\hat{p}}$ with

$$\mathbf{V}_h^{\hat{p}} = \{\mathbf{v} \in [L_2(\Omega)]^m : \mathbf{v}|_\kappa \circ \sigma_\kappa \in [\mathcal{Q}_{\hat{p}}(\hat{\kappa})]^m \,,\ \kappa \in \mathcal{T}_h\}, \tag{70}$$

and $\hat{p} > p$. Replacing the dual solution $\mathbf{z}$ in (67) by its approximation $\hat{\mathbf{z}}_h$ results in following approximate error representation formula

$$J(\mathbf{u}) - J(\mathbf{u}_h) \approx -\mathcal{N}(\mathbf{u}_h, \hat{\mathbf{z}}_h - \mathbf{z}_h) \equiv \sum_{\kappa \in \mathcal{T}_h} \hat{\eta}_\kappa, \tag{71}$$

and an analogous formula for the approximate error bound. We note that the error introduced into the error representation through this replacement consists of the linearization and the discretization error of the dual problem, see [20] for a more detailed discussion.

8.5 Adaptive mesh refinement

In this section we consider the design of an adaptive algorithm to ensure the efficient computation of the given target functional $J(\cdot)$ of practical interest. To this end, we employ the approximate error bound $\sum_{\kappa\in\mathcal{T}_h} |\hat{\eta}_\kappa|$ to determine when the desired level of accuracy has been achieved. For example, suppose that the aim of the computation is to compute $J(\cdot)$ such that the error $|J(\mathbf{u}) - J(\mathbf{u}_h)|$ is less than some user-defined tolerance `TOL`, i.e.,

$$|J(\mathbf{u}) - J(\mathbf{u}_h)| \leq \texttt{TOL};$$

then, in practice we may enforce the stopping criterion

$$\sum_{\kappa\in\mathcal{T}_h} |\hat{\eta}_\kappa| \leq \texttt{TOL}.$$

If this condition is not satisfied on the current finite element mesh $\mathcal{T}_h$, then the elementwise terms $\hat{\eta}^\kappa$ are employed as local error indicators to guide mesh refinement and coarsening. The cycle of the adaptive mesh refinement is outlined as follows:

1. Construct an initial mesh $\mathcal{T}_h$.
2. Compute $\mathbf{u}_h \in \mathbf{V}_h^p$ on the current mesh $\mathcal{T}_h$.
3. Compute $\hat{\mathbf{z}}_h \in \mathbf{V}_h^{\hat{p}}$, where $\mathbf{V}_h^{\hat{p}}$ is a finite element space based on the (same) computational mesh $\mathcal{T}_h$, but consisting of piecewise (discontinuous) polynomials of degree $\hat{p} > p$, see (70).
4. Evaluate the approximate a posteriori error bound $\sum_{\kappa\in\mathcal{T}_h} |\hat{\eta}_\kappa|$.
5. If $\sum_{\kappa\in\mathcal{T}_h} |\hat{\eta}_\kappa| \leq \texttt{TOL}$, where `TOL` is a given tolerance, then `STOP`.
6. Otherwise, refine and coarsen a fixed fraction of the total number of elements according to the size of $|\hat{\eta}_\kappa|$ and generate a new mesh $\mathcal{T}_h$; `GOTO` 2.

8.6 Numerical examples

In this section we give several examples which shall illustrate and explain the structure of dual (adjoint) solutions. In particular, we explain the dual solution's role associated with information transport, error transport as well as error accumulation in numerical simulations, which is a key ingredient of error estimation and goal-oriented adaptive mesh refinement. We show the adjoint solutions for various problems, we demonstrate the accuracy of the error estimation as well as the performance of the adaptive refinement algorithm.

In the first two examples we revisit standard test cases of inviscid flows, the Ringleb flow problem and the supersonic flow past a wedge. In order to track paths of information and error transport in these flows and to understand the structure of the dual solution and the resulting adaptive mesh refinement, here, we choose a particularly simple target quantity, namely the solution (one component of it only) at one specific point in the computational domain.

In the last example we show the application of this approach to aerodynamics. Here, we consider the accurate approximation of the aerodynamical force coefficients of a supersonic viscous flow around an airfoil.

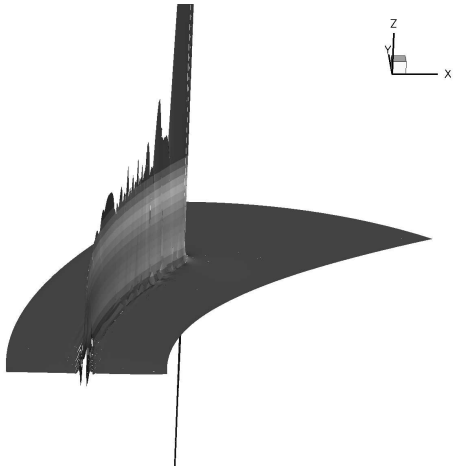


Figure 18: Ringleb flow problem, $J(u) = \rho(-0.4, 2)$. z_1 component of dual solution.

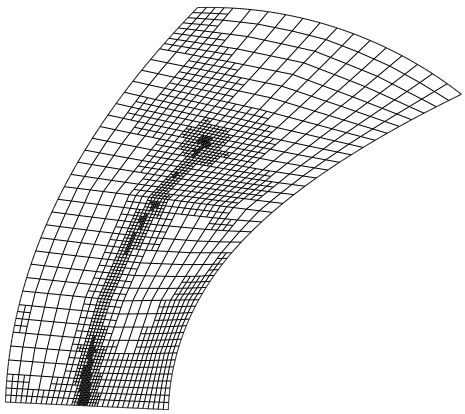


Figure 19: Ringleb flow problem, $J(u) = \rho(-0.4, 2)$. Mesh constructed using the indicators $\hat{\eta}_\kappa$.

8.6.1 Ringleb flow problem

As first example we consider the Ringleb flow problem already introduced in Section 3.1. We choose the target functional to be

$$J(u) = \rho(-0.4, 2).$$

We note that this target functional is singular in the sense that it leads to a considerably rough dual solution that mainly consists of a single spike transported in reverse direction of the flow, see Figure 18.

The mesh produced using the indicators $\hat{\eta}_\kappa$ is shown in Figure 19. Here, we see that the mesh is mostly concentrated in the neighbourhood of the characteristic upstream of the point of interest. However, due to the elliptic nature of the flow in the subsonic region, a circular region containing the point of interest is also refined, together with a strip of cells in the vicinity of the wall on the right–hand side of the domain enclosing the supersonic region of the flow.

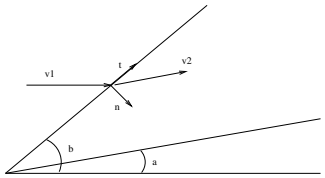


Figure 20: Geometry for the supersonic compression corner.

8.6.2 Supersonic flow past a wedge

In this example we study the formation of an oblique shock when a supersonic flow is deflected by a sharp object or wedge (also called *supersonic compression corner*). Here, we consider a Mach 3 flow, over a compression corner of angle α which results in the development of a shock at an angle β, cf. Figure 20. By employing the Rankine–Hugoniot jump conditions, the analytical solution to this problem for a given α may be determined, see [1, 34] among others. Here, we select the wedge angle $\alpha = 9.5°$; thereby, the angle of the shock is given by $\beta = 26.9308°$. Furthermore, the true solution on the left– and right–hand side of the shock, in terms of conservative variables $(\rho, \rho v_1, \rho v_2, \rho E)$, are given by

$$\mathbf{u}_{\text{left}} \approx \begin{pmatrix} 1 \\ 3.5496 \\ 0 \\ 8.8 \end{pmatrix} \quad \text{and} \quad \mathbf{u}_{\text{right}} \approx \begin{pmatrix} 1.6180 \\ 5.2933 \\ 0.8858 \\ 13.8692 \end{pmatrix},$$

respectively.

Again, for simplicity, we consider a point evaluation; In particular, the point value

$$J(\mathbf{u}) = \rho(5, 2.05)$$

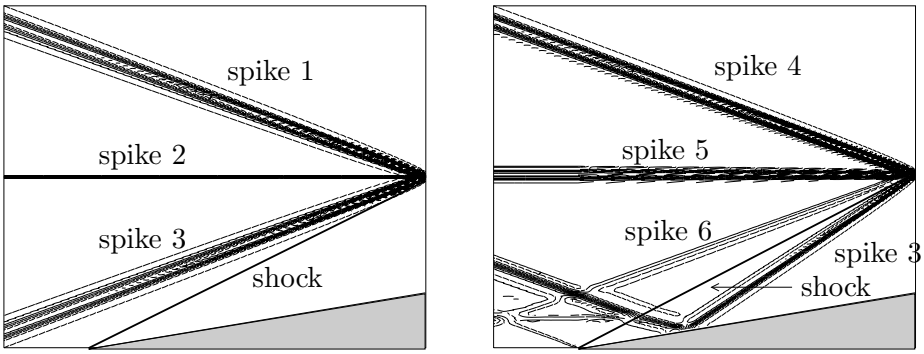
of the density just *in front of* the shock. In Figure 21(a) we show the z_1 component of the corresponding dual solution. It consists of three 'spikes', labelled 1, 2 and 3 in Figure 21(a), originating from the point of interest. These spikes are transported upstream along the characteristics corresponding to the three eigenvalues v and $v \pm c$, with $v = |\mathbf{v}| = \sqrt{v_1^2 + v_2^2}$ denoting the velocity of the gas and $c = \sqrt{\gamma p/\rho}$ the speed of sound. We note that the support of this dual solution does not intersect the region of the computational domain where the shock in the primal solution is located.

In the following, we consider the more interesting case of a point evaluation of the density

$$J(\mathbf{u}) = \rho(5, 2.01)$$

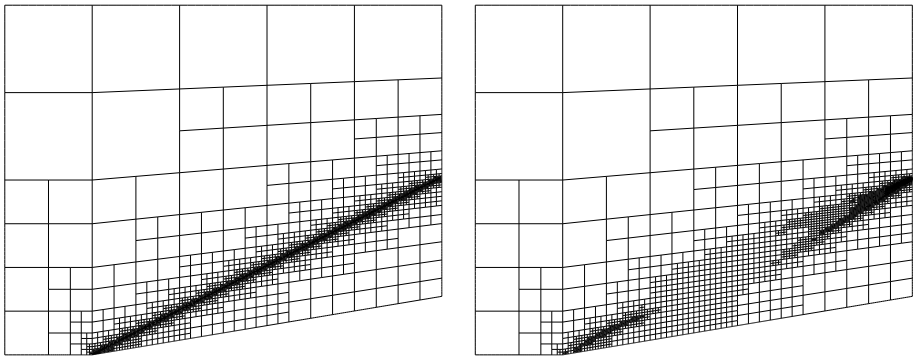
just *behind* the shock. Here, the support of the dual solution, see Figure 21(b), now intersects the region containing the shock and has a rather complicated structure. The two upper spikes of the dual solution both cross the shock in the neighbourhood of the point of evaluation. At their crossing points they again each split into a further three spikes. These six spikes correspond to the three pairs of spikes, labelled spikes 4, 5 and 6 in Figure 21(b); the two spikes in each pair cannot be distinguished on the resolution shown, as they are extremely close together. Spike 3, corresponding to the same spike in Figure 21(a), is reflected off the inclined wall and crosses the shock at its bottom part.

A comparison of the dual solution in Figure 21(b) and the mesh in Figure 22(a), produced by the indicators $\hat{\eta}_\kappa$, shows that the mesh has only been refined along the support



(a) (b)

Figure 21: Supersonic flow past a wedge: z_1 component of dual solution for the supersonic compression corner for point evaluation of the density (a) in front of shock (b) behind shock.

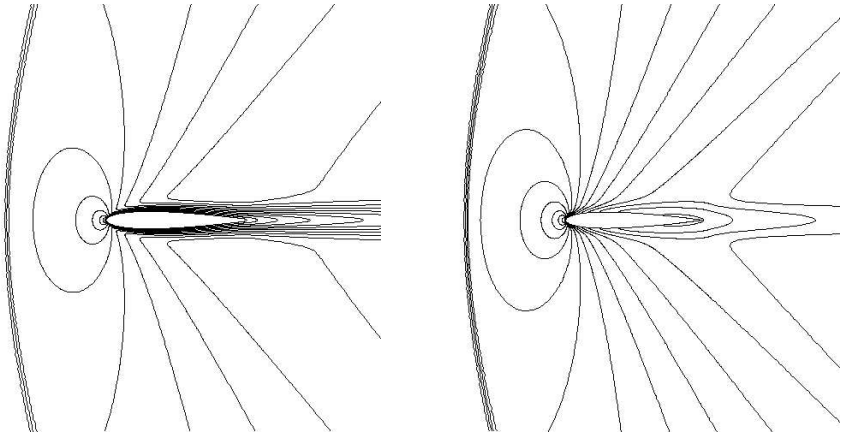


(a) (b)

Figure 22: Supersonic flow past a wedge: point evaluation of the density behind the shock (a) Mesh constructed using *ad hoc* error indicators with 3821 elements ($|J(u) - J(u_h)| = 8.938 \times 10^{-3}$) (b) Mesh constructed using the indicators $\hat{\eta}_\kappa$ with 3395 elements ($|J(u) - J(u_h)| = 2.888 \times 10^{-4}$);

of spikes 3 and 6 in the vicinity of the top part of the shock, and in the neighbourhood of the point where spike 3 crosses the bottom part of the shock. Comparing this mesh with the mesh in Figure 22(b) produced using traditional (or so-called *ad hoc*) indicators, we see that the adaptively refined meshes generated by employing the indicators $\hat{\eta}_\kappa$ are significantly more efficient than those produced using the traditional error indicators. Indeed, the true error in the computed functional is over an order of magnitude smaller when the indicators $\hat{\eta}_\kappa$ are employed.

This demonstrates that it is not necessary to refine the entire shock in this example to gain an accurate value of the target quantity under consideration, but only those parts that influence the target quantity either by material transport (eigenvalue v), or by information transported by the sound waves (eigenvalues $v \pm c$).



(a) (b)

Figure 23: $M = 1.2, \mathrm{Re} = 1000, \alpha = 0^\circ$ flow around the NACA0012 airfoil: (a) Mach isolines and (b) density isolines.

8.6.3 Supersonic flow around an airfoil

In this subsection we present a numerical example demonstrating that the approximate error representation $-\mathcal{N}(\mathbf{u}_h, \hat{\mathbf{z}}_h - \mathbf{z}_h) = \sum_{\kappa \in \mathcal{T}_h} \hat{\eta}_\kappa$, cf. (71), which was derived from the (exact) error representation (68) by replacing the dual solution $\mathbf{z}$ by an approximate dual solution $\hat{\mathbf{z}}_h$, gives a good approximation to the true error meassured in terms of the specific target quantity $J(\mathbf{u})$. Furthermore, we highlight the advantages of designing an adaptive finite element algorithm as outlined in Section 8.5, based on *adjoint-based* (or *dual-weighted residual*) indicators $\hat{\eta}_\kappa$, in comparison to *residual-based* indicators, cf. [24], which do not require the solution of an auxiliary (dual) problem.

To this end, we consider a horizontal viscous flow at $M = 1.2$ and $\mathrm{Re} = 1000$, with an adiabatic no-slip boundary condition imposed on the profile. Due to the Reynold's number which is higher than in the test case in Section 7.4.2 the bow shock of the flow, see Figure 23, is sharper and, due to the lower Mach number, it is located at a larger distance in front of the airfoil than in Section 7.4.2. Furthermore, there are two weak shocks emanating from the trailing edge of the airfoil, see Figure 24.

In this example, the target quantity $J(\mathbf{u})$ is chosen to be an aerodynamical force coefficient, see Section 6.1. As this flow is symmetric about the x-axis, both lift coefficients, c_{lp} and c_{lf}, vanish. On the basis of fine grid computations the reference values of the pressure induced drag, c_{dp}, and the viscous drag, c_{df}, are given by $J_{c_{\mathrm{dp}}}(\mathbf{u}) \approx 0.10109$ and $J_{c_{\mathrm{df}}}(\mathbf{u}) \approx 0.10773$, respectively.

In the following, we consider the approximation of the pressure induced drag, c_{dp}, i.e. the target quantity is $J(\cdot) = J_{c_{\mathrm{dp}}}(\cdot)$. In Table 7, we collect the data of the adaptive algorithm based on employing the adjoint-based indicators $\hat{\eta}_\kappa$. Here, we show the number of elements and degrees of freedom (DoF) for $p = 1$ (bilinear elements), the true error in the target quantity, $J(\mathbf{u}) - J(\mathbf{u}_h)$, the approximate error representation formula $\sum_{\kappa \in \mathcal{T}_h} \hat{\eta}_\kappa$ and the effectivity index $\theta = \sum_{\kappa \in \mathcal{T}_h} \hat{\eta}_\kappa / (J(\mathbf{u}) - J(\mathbf{u}_h))$ of the error estimation. First, we note that on all meshes the right sign of the error is predicted, which is always negative in this computation, i.e. the computed c_{dp} values converge to the reference value from above. Furthermore, from the second mesh onwards, the approximate error representations represent a very good approximation to the true errors, also indicated by the effectivity indices θ being very close to one.

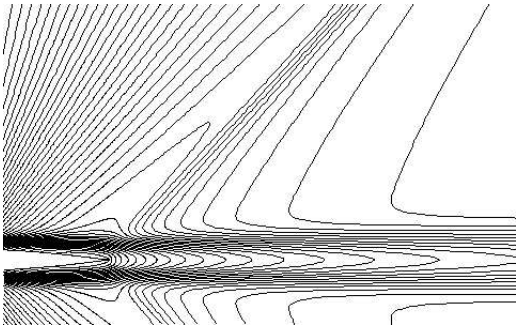
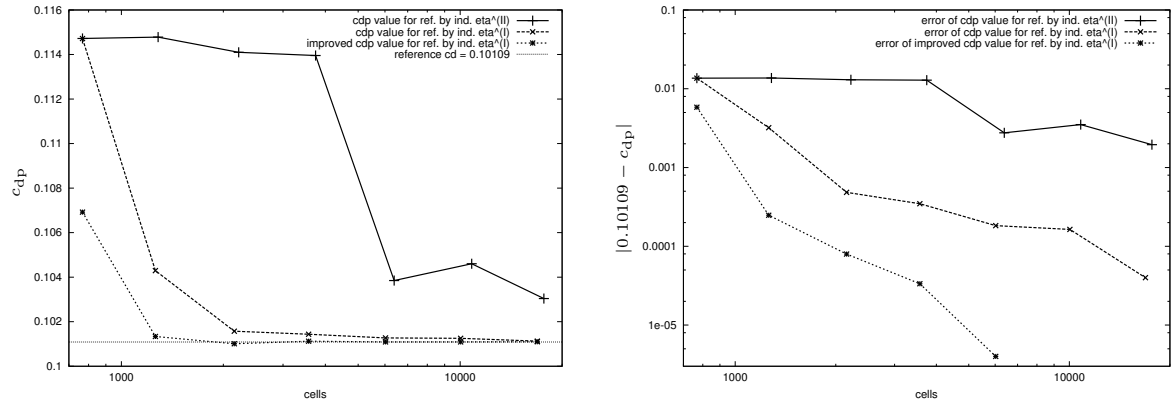


Figure 24: $M = 1.2, \mathrm{Re} = 1000, \alpha = 0°$ flow around the NACA0012 airfoil: Zoom of density isolines at trailing edge.

# el.	# DoFs	$J(\mathbf{u}) - J(\mathbf{u}_h)$	$\sum_\kappa \hat{\eta}_\kappa$	θ
768	12288	-1.363e-02	-6.312e-03	0.46
1260	20160	-3.203e-03	-2.995e-03	0.94
2154	34464	-4.844e-04	-5.368e-04	1.11
3570	57120	-3.474e-04	-3.333e-04	0.96
6021	96336	-1.835e-04	-1.856e-04	1.01
10038	160608	-1.644e-04	-1.653e-04	1.01

Table 7: Viscous $M = 1.2, \mathrm{Re} = 1000, \alpha = 0°$ flow around the NACA0012 airfoil: Adaptive algorithm for the accurate approximation of c_{dp}.



(a) (b)

Figure 25: $M = 1.2, \mathrm{Re} = 1000, \alpha = 0°$ flow around the NACA0012 airfoil: (a) $J_{c_{\mathrm{dp}}}(\mathbf{u})$ values on adaptive refined meshes using indicator $\eta_\kappa^{(\mathrm{II})}$, $J_{c_{\mathrm{dp}}}(\mathbf{u})$ and the improved values, $\tilde{J}_{c_{\mathrm{dp}}}(\mathbf{u}) = J_{c_{\mathrm{dp}}}(\mathbf{u}) + \sum_\kappa \hat{\eta}_\kappa$, on adaptive refined meshes using indicator $\hat{\eta}_\kappa^{(\mathrm{I})}$ versus number of elements; (b) Error of these values versus number of elements.

In Figure 25 we compare the true error in the target quantity for the two mesh refinement strategies based on the adjoint-based indicator $\hat{\eta}_\kappa$ and on the residual-based indicator, respectively. We see, that on the first three refinement steps when employing the residual-based indicator the accuracy in the target quantity is hardly improved. In contrast to that, when using adjoint-based indicators, the error decreases significantly faster, being a factor of more than three smaller already after the second refinement step than the error on the finest residual-based refined mesh. Furthermore, the computed values of the target quantity $J(\mathbf{u}_h)$ can be enhanced by employing the approximate error representation $\sum_{\kappa\in\mathcal{T}_h} \hat{\eta}_\kappa$ to yield an *improved value* of the target quantity, $\tilde{J}(\mathbf{u}_h) = J(\mathbf{u}_h) + \sum_{\kappa\in\mathcal{T}_h} \hat{\eta}_\kappa$. In Figure 25 we see, that the improved values, $\tilde{J}(\mathbf{u}_h)$, are significantly more accurate than the (baseline) $J(\mathbf{u}_h)$ values, and even show a higher rate of convergence. In fact, it can be shown, see [24], that this value has a higher order of convergence than $J(\mathbf{u}_h)$, provided the primal and the dual solutions are smooth and the dual solution is approximated using higher-order polynomials. Furthermore, the approximate error representation is close to the true error even in cases of smooth dual solutions but possibly non-smooth primal solutions, see [20] for a more detailed discussion.

The large difference in the performance, see Figure 25, of the adjoint-based indicators and the residual-based indicators in producing adaptively refined meshes for the accurate approximation of the target quantity c_{dp}, is due to the very different parts of the computational meshes being marked for refinement by the two types of indicators. Figures 26 (a) & (b) show the finest mesh produced by employing the residual-based indicator. We see, that this refinement criterion aims at resolving all flow features: the extensive bow shock, the wake of the flow behind the airfoil as well as the weak shocks emanating from the trailing edge of the airfoil. In contrast to that, the refinement of the mesh produced by employing the adjoint-based indicator, see Figures 26 (c) & (d), is very concentrated close to the airfoil. In particular, the bow shock is mainly resolved in a small region upstream of the profile only, and there is no refinement at all at the position of the bow shock beyond six chord lengths above and below the profile. Furthermore, the weak shocks emanating from the trailing edge are not resolved and there is no refinement in the wake of the flow beyond three chord lengths behind the profile. Instead, the refinement of the mesh is concentrated near the leading edge of the profile and in the boundary layer of the flow. All other parts of the computational domain are recognized by the adjoint-based indicator to be of minor importance for the accuracy of the c_{dp} target quantity. In fact, the dual (adjoint) solution, see Figures 27 and 28, includes the crucial information concerning which local residuals contribute to the error in the target quantity and to what extent. Herewith, it offers all necessary information of error transport and accumulation. Finally, the adjoint-based indicators mark only those parts of the domain for refinement where residuals of the flow solution significantly contribute to the error of the target quantity, i.e. all parts which are important for the accurate approximation of the target quantity.

9 Acknowledgements

All computations have been performed using a DG flow solver based on the `deal.II` library [4].

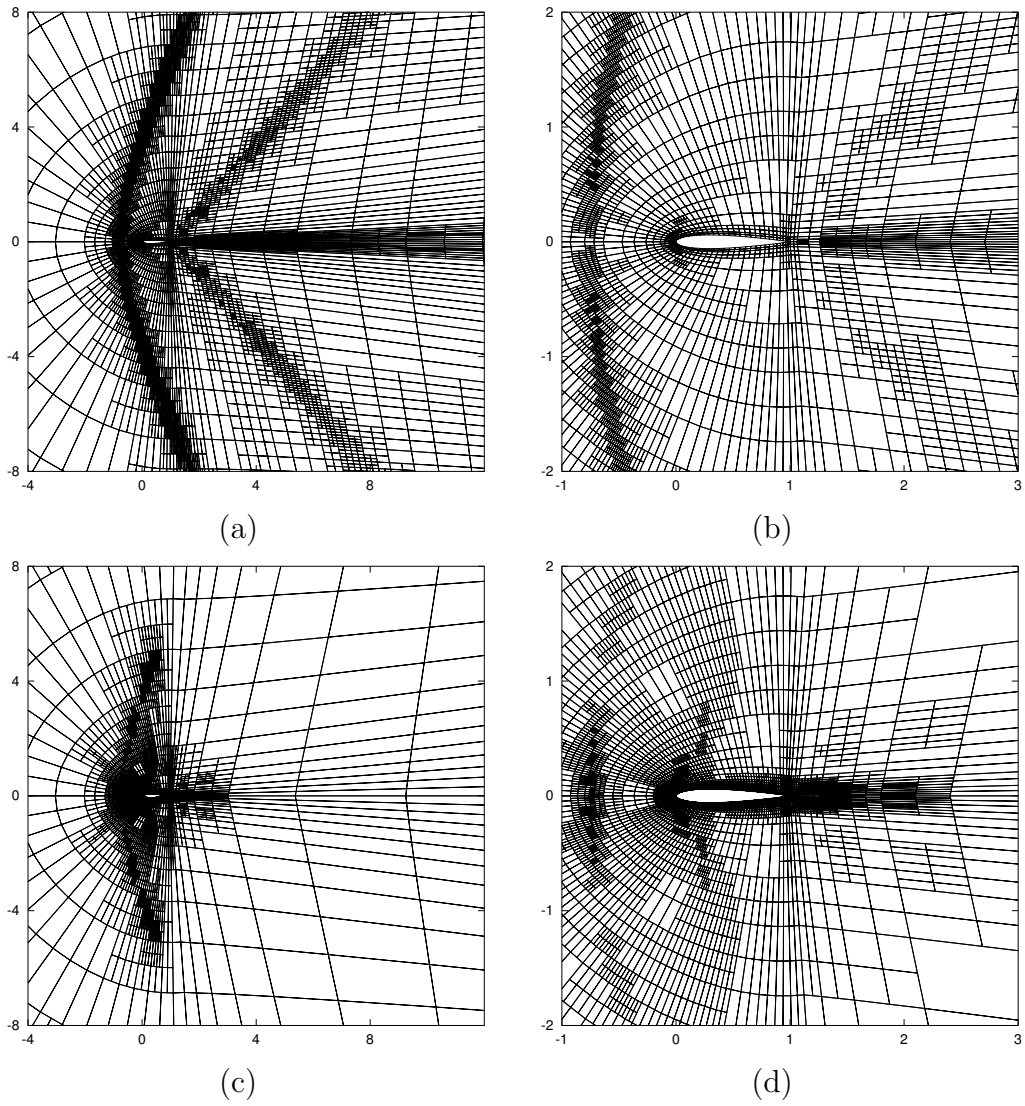


Figure 26: $M = 1.2, \mathrm{Re} = 1000, \alpha = 0°$ flow around the NACA0012 airfoil: (a) & (b) residual-based refined mesh of 17670 elements with 282720 degrees of freedom and $|J_{c_{\mathrm{dp}}}(\mathbf{u}) - J_{c_{\mathrm{dp}}}(\mathbf{u}_h)| = 1.9 \cdot 10^{-3}$; (c) & (d) adjoint-based refined mesh for c_{dp}: mesh of 10038 elements with 160608 degrees of freedom and $|J_{c_{\mathrm{dp}}}(\mathbf{u}) - J_{c_{\mathrm{dp}}}(\mathbf{u}_h)| = 1.6 \cdot 10^{-4}$.

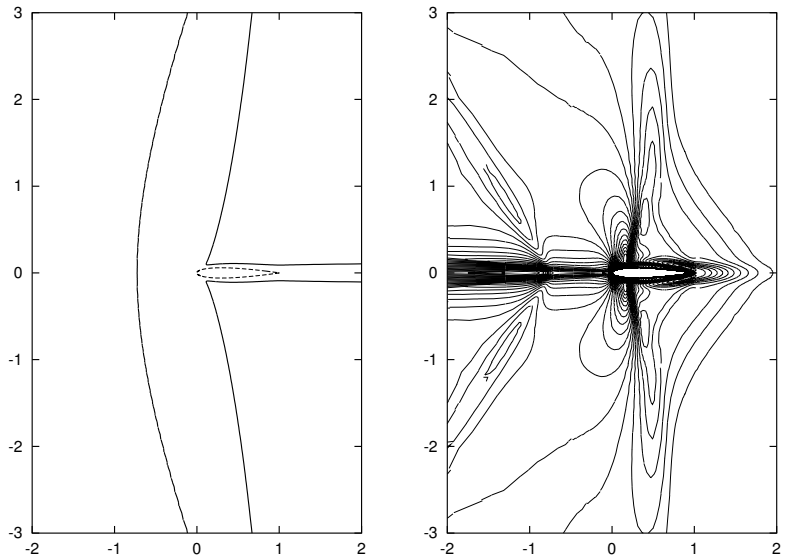


Figure 27: Viscous flow at $M = 1.2, \mathrm{Re} = 1000, \alpha = 0°$ around the NACA0012 airfoil: (a) Sonic isolines of the flow solution; (b) isolines of the $\hat{z}_1$ component of the computed adjoint solution $\hat{\mathbf{z}}$.

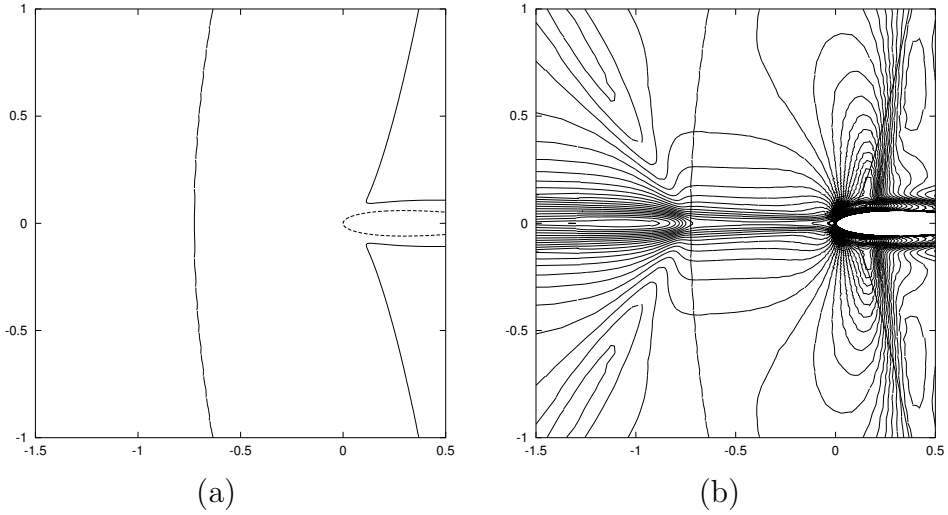


Figure 28: Viscous flow at $M = 1.2, \mathrm{Re} = 1000, \alpha = 0°$ around the NACA0012 airfoil: Zoom of (a) sonic ($M = 1$) isolines of the flow solution; (b) together with $\hat{z}_1$ isolines.

References

[1] J. D. Anderson, editor. *Fundamentals of Aerodynamics*. McGraw-Hill, 3rd edition, 2001.

[2] D. Arnold, F. Brezzi, B. Cockburn, and D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39(5):1749–1779, 2002.

[3] S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. *PETSc Web page*, 2001. http://www.mcs.anl.gov/petsc.

[4] W. Bangerth, R. Hartmann, and G. Kanschat. `deal.II` *Differential Equations Analysis Library, Technical Reference*. http://www.dealii.org/, 5.1 edition, Dec. 2004. First edition 1999.

[5] F. Bassi and S. Rebay. Accurate 2d Euler computations by means of a high order discontinuous finite element method. In *XIVth ICNMFD, Bangalore, July 11-15, 1994*, Lecture Notes in Physics. Springer, 1996.

[6] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *J. Comp. Phys.*, 131:267–279, 1997.

[7] F. Bassi and S. Rebay. High-order accurate discontinuous finite element solution of the 2d Euler equations. *J. Comp. Phys.*, 138:251–285, 1997.

[8] F. Bassi, S. Rebay, G. Mariotti, S. Pedinotti, and M. Savini. A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows. In R. Decuypere and G. Dibelius, editors, *2nd European Conference on Turbomachinery Fluid Dynamics and Thermodynamics, Antwerpen, Belgium, March 5–7, 1997*, pages 99–108. Technologisch Instituut, 1997.

[9] C. Baumann and J. Oden. A discontinuous *hp* finite element method for the Euler and Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 31:79–95, 1999.

[10] C. Baumann and J. Oden. An adaptive-order discontinuous Galerkin method for the solution of the Euler equations of gas dynamics. *International Journal for Numerical Methods in Engineering*, 47:61–73, 2000.

[11] R. Becker and R. Rannacher. A feed-back approach to error control in finite element methods: Basic analysis and examples. *East–West J. Numer. Math.*, 4:237–264, 1996.

[12] R. Becker and R. Rannacher. An optimal control approach to error estimation and mesh adaptation in finite element methods. *Acta Numerica*, 10:1–102, 2001.

[13] M. O. Bristeau, R. Glowinski, J. Periaux, and H. Viviand. Numerical simulation of compressible Navier-Stokes flows. In *Numerical Simulation of Compressible Navier-Stokes flows*, volume 18 of *Notes on Numerical Fluid Mechanics*. Vieweg-Verlag, 1986. Proceedings of the GAMM workshop held at INRIA, France, Dec. 4-6, 1985.

[14] G. Chiocchia. Exact solutions to transonic and supersonic flows. Technical Report AR-211, AGARD, 1985.

[15] B. Cockburn and C.-W. Shu. The Runge–Kutta discontinuous Galerkin finite element method for conservation laws V: Multidimensional systems. *J. Comput. Phys.*, 141:199–224, 1998.

[16] V. Dolejsi. On the discontinuous Galerkin method for the numerical solution of the Navier-Stokes equations. *Int. J. Numer. Meth. Fluids*, 45:1083–1106, 2004.

[17] E. H. Georgoulis. hp-version interior penalty discontinuous galerkin finite element methods on anisotropic meshes. *IJNAM*, 3(1):52–79, 2006.

[18] R. Hartmann. Adaptive FE Methods for Conservation Equations. In H. Freistühler and G. Warnecke, editors, *Hyperbolic Problems: theory, numerics, applications: eighth international conference in Magdeburg, February, March 2000*, volume 141 of *International series of numerical mathematics*, pages 495–503. Birkhäuser, Basel, 2001.

[19] R. Hartmann. *Adaptive Finite Element Methods for the Compressible Euler Equations*. PhD thesis, University of Heidelberg, 2002.

[20] R. Hartmann. Adaptive discontinuous Galerkin methods with shock-capturing for the compressible Navier-Stokes equations. *Int. J. Numer. Meth. Fluids*, 2006. To appear.

[21] R. Hartmann and P. Houston. Adaptive discontinuous Galerkin finite element methods for nonlinear hyperbolic conservation laws. *SIAM J. Sci. Comp.*, 24:979–1004, 2002.

[22] R. Hartmann and P. Houston. Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *J. Comp. Phys.*, 183:508–532, 2002.

[23] R. Hartmann and P. Houston. Symmetric interior penalty DG methods for the compressible Navier–Stokes equations I: Method formulation. *Int. J. Num. Anal. Model.*, 3(1):1–20, 2006.

[24] R. Hartmann and P. Houston. Symmetric interior penalty DG methods for the compressible Navier–Stokes equations II: Goal–oriented a posteriori error estimation. *Int. J. Num. Anal. Model.*, 3(2):141–162, 2006.

[25] P. Houston, C. Schwab, and E. Süli. Stabilized *hp*-finite element methods for first–order hyperbolic problems. *SIAM J. Numer. Anal.*, 37(5):1618–1643, 2000.

[26] J. Jaffre, C. Johnson, and A. Szepessy. Convergence of the discontinuous Galerkin finite element method for hyperbolic conservation laws. *Math. Models and Methods in Appl. Sciences*, 5:367–386, 1995.

[27] C. Johnson and J. Pitkäranta. An analysis of the discontinuous Galerkin method for a scalar hyperbolic equ ation. *Math. Comp.*, 46(173):1–26, 1986.

[28] D. Kröner. *Numerical Schemes for Conservation Laws*. Wiley-Teubner, 1997.

[29] P. LeSaint and P. A. Raviart. On a finite element method for solving the neutron transport equation. In C. de Boor, editor, *Mathematical Aspects of Finite Elements in Partial Differential Equations*, pages 89–145. Academic Press, 1974.

[30] T. E. Peterson. A note on the convergence of the discontinuous Galerkin method for a scalar hyperbolic equation. *SIAM J. Numer. Anal.*, 28(1):133–140, 1991.

[31] W. H. Reed and T. R. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.

[32] G. Richter. An optimal–order error estimate for the discontinuous Galerkin method. *Math. Comp.*, 50:75–88, 1988.

[33] H. Schlichting and K. Gersten. *Boundary-Layer Theory.* Springer, 2003.

[34] H. Schlichting and E. Truckenbrodt. *Aerodynamics of the Airplane*, volume 1. McGraw-Hill, 1979.

[35] E. F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics.* Springer, 1997.

[36] J. van der Vegt and H. van der Ven. Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows, I. General formulation. *J. Comp. Phys.*, 182:546–585, 2002.